Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6641180, 13 pages
https://doi.org/10.1155/2021/6641180

Research Article

A Folded Concave Penalty Regularized Subspace Clustering Method to Integrate Affinity and Clustering

Wenjuan Zhang,[1] Xiangchu Feng,[2] Feng Xiao,[3] and Yunmei Chen[4]

[1]*School of Science, Xi'an Technological University, Xi'an 710021, Shaanxi, China*
[2]*School of Mathematics and Statistics, Xidian University, Xi'an 710071, Shaanxi, China*
[3]*School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, Shaanxi, China*
[4]*Department of Mathematics, University of Florida, Gainesville 32611, FL, USA*

Correspondence should be addressed to Feng Xiao; xffriends@163.com

Received 22 October 2020; Accepted 26 April 2021; Published 17 May 2021

Academic Editor: Xingbao Gao

Most sparse or low-rank-based subspace clustering methods divide the processes of getting the affinity matrix and the final clustering result into two independent steps. We propose to integrate the affinity matrix and the data labels into a minimization model. Thus, they can interact and promote each other and finally improve clustering performance. Furthermore, the block diagonal structure of the representation matrix is most preferred for subspace clustering. We define a folded concave penalty (FCP) based norm to approximate rank function and apply it to the combination of label matrix and representation vector. This FCP-based regularization term can enforce the block diagonal structure of the representation matrix effectively. We minimize the difference of l_1 norm and l_2 norm of the label vector to make it have only one nonzero element since one data only belong to one subspace. The index of that nonzero element is associated with the subspace from which the data come and can be determined by a variant of graph Laplacian regularization. We conduct experiments on several popular datasets. The results show our method has better clustering results than several state-of-the-art methods.

1. Introduction

In machine learning and data mining, clustering is one of the most important topics in unsupervised learning. Given a set of data points, clustering is to partition these points into several groups, with each of which called a cluster, such that data points in the same group have higher similarities than those in different groups. In the past decades, an enormous number of clustering algorithms have been proposed, such as K-means and its variants [1, 2], spectral clustering [3], nonnegative matrix factorization (NMF) [4], and subspace clustering [5]. In this paper, we focus on the methods of subspace clustering.

Contemporary is the era of high-dimensional data explosion. However, there is redundancy information included in those high-dimensional data so that their intrinsic dimension is often much smaller. In many computer vision and machine learning problems, one often assumes that the data are drawn from a union of multiple low-dimensional linear subspaces. Thus, subspace clustering of such data has been studied extensively. Nowadays, many works proposed the assumption of linear subspace may not always be true for many real high-dimensional data. The proposed data may be better modeled by nonlinear manifolds [6–8]. The common strategy employed in these works is to use the logarithm mapping projecting data onto the tangent space at each data point which is a linear space, so that all the strategies applicable to linear space can be used. These methods finally still need to establish a model in linear space. Therefore, research of modeling methods in linear space is still essential. The existing methods for subspace clustering can be roughly divided into four groups: statistical learning-based methods, factorization-based methods, algebra-based

methods, and sparsity-based methods (e.g., Sparse Subspace Clustering (SSC) [9] and Low-Rank Representation (LRR)) [10]. In this paper, we focus on the fourth group.

In recent years, a lot of methods basing on deep learning [11] have been proposed. These methods obtained extremely competitive results in many fields on image and computer vision. A discussion hot point in computational imaging is if it is time to discard the classic methods and fully replace them with deep learning-based methods. On the one hand, a prerequisite for deep learning-based methods is a huge number of samples. However, there exist some situations where no data or only a small number of data can be obtained. In this case, the knowledge-based modeling methods are more suitable. On the other hand, classic methods have a clear structure and theoretical guarantee. They are based on the knowledge of the problem we are trying to solve rather than seeking for best performance by intuitively choosing architectures or trial and error. To the best of our knowledge, most of the existing learning-based clustering methods only learn some autoencoder features [12]. The final clustering is still obtained by applying k-means or SSC. Another challenge for deep learning is that a clustering model trained from some datasets may not be effective for other sets, but classic methods have better generalization ability.

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in R^{m\times n}\,(m < n)$ be a collection of m-dimensional data drawn from the union of linear subspaces $\{S_i\}_{i=1}^{c}$. Each S_i includes n_i data of $\mathbf{X}$. The task of subspace clustering is to cluster the data in $\mathbf{X}$ according to those independent subspaces. For each $j \in \{1, 2, \ldots, n\}$, consider the data $\mathbf{x}_j$ as a linear combination of all data in $\mathbf{X}$, i.e., $\mathbf{x}_j = \mathbf{X}\mathbf{z}_j$, here $\mathbf{z}_j = (Z_{kj})_{k=1}^{n}$ is called a representation vector, we assume $Z_{jj} = 0$ to avoid the trivial solution. Suppose $\mathbf{x}_j \in S_i\,(i \in \{1, 2, \ldots, c\})$, then only the coefficients associated with the data from S_i are not zero, and the others are all zero. Assuming $A_j = \{k\colon Z_{kj} \neq 0\}$, we want to find the solution to the following problem:

$$\arg\min_{\left\{z_j \in R^n : Z_{kj}=0\left(k\in A_j^c\right), Z_{jj}=0\right\}} \left\|\mathbf{x}_j - \mathbf{X}\mathbf{z}_j\right\|_2^2, \quad (j = 1, 2, \ldots, n). \tag{1}$$

which means to find the oracle solution [9].

According to the above statement, there must be at least one matrix $\mathbf{Z} = (Z_{ij})_{n\times n} \in R^{n\times n}$ satisfying

$$\mathbf{X} = \mathbf{X}\mathbf{Z}, \quad \text{s.t. } Z_{jj} = 0\,(j = 1, 2, \ldots, n). \tag{2}$$

Equation (2) actually has an infinite amount of solutions. Any solution is called a representation matrix. When handling 2D data, with each data being a matrix, [13] shows the strategy of converting all data to vectors severely damages inherent structural information and correlations of the original data. They proposed to learn a 2D projection matrix such that the most expressive structural information is retained in the spanned subspace. In our work, we still consider the vectorized data for simplicity since our method is also suitable for the projected vectorized data, and it is not within the scope of this paper to discuss how to deal with 2D data.

A good representation matrix should have the properties of sparsity between subspaces and density within a subspace, which means each query data $\mathbf{x}_j$ is represented by a small number of subspaces, and once one subspace is selected, it is in favor of using more data from the same subspace. The work of [14] introduced a family of new regression models and estimated a representation model while accounting for discriminativeness between clusters. Here we achieve this property of the representation matrix by forcing it to have block diagonal structure since the block diagonal structure of $\mathbf{Z}$ directly induces a segmentation of the data (each block corresponds to a cluster). Reference [15] stated that under the ideal conditions, i.e., the data are noiseless and the subspaces are independent (i.e., none of the subspaces can be represented by other subspaces), as long as the regularizer for $\mathbf{Z}$ satisfies the EBD (Enforced Block Diagonal) conditions, and the optimal representation matrix is block diagonal. However, as $\mathbf{X}$ contains noise, which is inevitable in any application, $\mathbf{Z}$ may not be a strict block diagonal matrix. Therefore, it is difficult to decide how large the representation coefficient between two data should be to group them into the same subspace. Usually, many previous subspace clustering methods are to find a matrix $\mathbf{Z}$ firstly; then using $(|\mathbf{Z}| + |\mathbf{Z}'|/2)$ as an affinity matrix, the spectral clustering such as normalized cuts can be applied to get the subspace clustering result, just as the classic SSC and LRR and many other variants thereafter have done. All these methods divide the solution of the representation matrix and the final clustering result into two independent steps. We propose to integrate the affinity matrix and the data labels into a model to make them interact and promote each other and finally improve the clustering performance.

Furthermore, in terms of the penalty for representation matrix $\mathbf{Z}$, most methods based on SSC and LRR seek $\mathbf{Z}$ with the most sparse or lowest rank constraint. In fact, for subspace clustering, it is more important for $\mathbf{Z}$ to have block diagonal structure than to be the most sparse or lowest rank matrix. We give a low-rank-based regularization term to enforce the block diagonal structure of $\mathbf{Z}$ directly. This regularization term is applied to a combination of the semisupervised label matrix and representation vector $\mathbf{z}_j$ other than $\mathbf{Z}$. Note that the rank function of a matrix is the l_0-norm of the singular value vector and solving such a l_0 minimization problem is usually difficult or even NP-hard. The standard approach is to replace the rank function with the convex nuclear norm [16, 17]. It has been proved that under certain incoherence assumptions on the singular values of the matrix, solving the convex nuclear norm regularized problem leads to a near-optimal low-rank solution [18]. On the other hand, it is pointed out that the nuclear norm is not accurate for rank approximation. Recent works develop various more accurate approximations to the rank function, such as the log-determinant rank approximation [19, 20], which significantly improves the learning performance. In addition, the nuclear norm of a matrix is the l_1-norm of the singular

values vectors. Fan and Li pointed out that l_1-norm penalty overpenalizes large entries of vectors; therefore, nuclear norm overpenalizes large singular values. Moreover, they proposed three criteria for good penalty functions [21]: unbiasedness, sparsity, and continuity at the origin. The l_1-norm satisfies both sparsity and continuity, but it is biased.

Recently, nonconvex penalties have drawn more and more attention to sparse learning problems because people believe that one of the possible solutions of nonconvex penalization problem could overcome the drawbacks of the unique solution of convex penalization problem. As a common practice, the l_1-norm can be replaced by the l_q-norm with $0 < q < 1$ if a more sparse solution is expected to be obtained [22, 23]. However, no theoretical guarantee with l_q-norm is made for reducing the modeling bias of l_1-norm. Based on those three properties proposed by Fan and Li, they proposed a new penalty function called the smoothly clipped absolute deviation penalty (SCAD) [21]. Recently, Zhang proved that a so-called min-max concave plus (MCP) penalty [24] also possesses three properties and achieves better performance than SCAD. Both SCAD and MCP are nonconvex and nearly unbiased. Extensive experiments [21, 24–30] have demonstrated the superiority of SCAD and MCP over the l_1-norm penalty. Furthermore, folded concave penalty (FCP) methods, including SCAD and MCP, have been shown to have strong oracle property for high-dimensional sparse estimation. As described above, we do expect to get the oracle solution of problem (1).

In this paper, a FCP regularized subspace clustering model is presented as follows:

$$\min_{\mathbf{g}_j\,(j\in P\setminus S)\in\Delta_+,\mathbf{Z}} \left\{ \begin{array}{l} \dfrac{1}{2}\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2 + \alpha \displaystyle\sum_{j\in P\setminus S}\left(\|\mathbf{g}_j\|_1 - \|\mathbf{g}_j\|_2\right) + \left\{ \dfrac{\beta}{2}\displaystyle\sum_{\substack{i\in PS \\ \text{or } j\in P\setminus S}} \|\mathbf{g}_i-\mathbf{g}_j\|_2^2\left(|Z_{ij}|+|Z_{ji}|\right) \right. \\ \left. +\dfrac{1}{2}\displaystyle\sum_{i\in S}\sum_{j\in S}\|\mathbf{g}_i-\mathbf{g}_j\|_2^{2p}\left(|Z_{ij}|+|Z_{ji}|\right)\right\} + \gamma\displaystyle\sum_{j\in P}\left\|\mathbf{G}\mathrm{Diag}\left(\mathbf{z}_j\right)\right\|_{P_{\lambda,a}} \end{array} \right\}. \tag{3}$$

We define a subspace dependent label matrix $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n] \in R^{c\times n}$, where $\mathbf{g}_j = (g_{1j}, g_{2j}, \ldots, g_{cj})^T \in R^c$ $(j = 1, 2, \ldots, n)$ is a label vector for data $\mathbf{x}_j$; here, c is the number of subspaces and n is the number of data. Since we do not know c, the dimension of $\mathbf{g}_j$ only needs to be set larger than c and at most equals to $\min\{m, n\}$. Here, we still use c to denote the dimension of $\mathbf{g}_j$. In our work, we mainly discuss the semisupervised case; i.e., some data already have labels and the others do not. Index set for all data and the labeled data are denoted as P and S separately; then $P\setminus S$ represents index set for the unlabeled data. For every $j \in S$, we assume $g_{ij} = 1$ when $\mathbf{x}_j$ is in ith class, while $g_{ij} = 0$ otherwise. The label vectors for $j \in P\setminus S$ need to be solved, and the set of all the unknown labels is defined as $\Delta_+ = \{\mathbf{g}|\mathbf{g} = (g_k)_{k=1}^c, g_k \geq 0, \sum_{k=1}^c g_k = 1\}$. Encouraged by the good properties of the FCP penalty, we give FCP-based norm $\|\cdot\|_{P_{\lambda,a}}$ to approximate rank function. The last term of our model (3) is to obtain rank minimization for matrix multiplication $\mathbf{G}$ $\mathrm{Diag}(\mathbf{z}_j)$. α, β, and γ are parameters used to balance roles of the three regularization terms. It is obvious that the larger the parameter, the more important the corresponding term in the minimization problem.

We give three regularization terms. The first one is minimizing the difference of l_1 norm and l_2 norm of the label vector $\mathbf{g}_j$ to make it have only one nonzero element. The clustering result for each data is induced by the index of that nonzero element. The second one is a variant of the graph Laplace regularization, which captures the nonlinear structures of the data. This term makes sure the data from the same subspace have the same label as much as possible. Therefore, the nonzero element of each label vector can properly correspond to the subspace from which the data is drawn. We apply a low-rank constraint to the combination of the label matrix and representation vector, as indicated in the third regularizer, to enforce $\mathbf{Z}$ to better satisfy block diagonal structure. In our work, the clustering result, namely the label for each data, is directly solved from the model rather than using spectral clustering methods. The labels and the representation matrix are contained in two regularization terms. The first part of the third term of (3) makes the nonzero element of each label vector accurately correspond to the subspace from which each query data comes if the representation matrix $\mathbf{Z}$ has the block diagonal structure. Vice versa, if each label has only one nonzero element which is associated with the accurate subspace, the second part of the third term and the fourth term make $\mathbf{Z}$ better meet the block diagonal structure. Therefore, the labels and the representation matrix interact and promote each other during the whole computing process.

The problem can be solved by using the Alternating Direction Method of Multiplier (ADMM) framework. The resulting nonconvex FCP minimization problem can be solved by the Linear Local Approximation (LLA) method [23], which solves the problem by minimizing a surrogate function that upper bounds the objective functional. The surrogate function is constructed by linearizing the penalty function. LLA guarantees to decrease the objective function value in each iteration. Due to the nonconvex of the FCP

problem, there are usually multiple local solutions, and the oracle property is established only for one of the local solutions. Breheny and Huang [25] have shown that with a Lasso-based initialization, LLA can avoid local maxima and saddle points, and with a high probability, an oracle solution can be obtained by using one-step LLA. In addition, once the oracle estimator is obtained, the LLA algorithm converges; namely, it produces the same estimator in the next iteration. We use the result of singular value thresholding as the initialization, which corresponds to the Lasso-based solution when applying to the nuclear norm minimization problem.

Some notations used in this work are defined as follows. For a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$ and $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ are l_1 norm and l_2 norm, respectively. $\mathrm{Diag}(\mathbf{x})$ is a diagonal matrix with diagonal entries being $\mathbf{x}$. For a matrix $\mathbf{X} = (X_{ij})$, $\|\mathbf{X}\|_F = \sqrt{\sum |X_{ij}|^2}$ denotes the Frobenius norm, $\|\mathbf{X}\|_* = \sum \sigma_i(\mathbf{X})$ is the nuclear norm, and here, $\sigma_i(\mathbf{X})$ is the ith singular value of $\mathbf{X}$. $|\mathbf{X}|$ represents the element-wise absolute value of $\mathbf{X}$. $\mathbf{X}^T$ denotes the transpose of $\mathbf{X}$. $\mathrm{Diag}(\mathbf{X})$ describes the diagonal matrix with diagonal components being X_{ii}. $\mathrm{diag}(\mathbf{X})$ is a vector with entries X_{ii}. $\mathrm{tr}(\mathbf{X})$ is the trace of the square matrix $\mathbf{X}$. $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and the zero matrix. $\mathbf{X} \geq \mathbf{0}$ means all entries of $\mathbf{X}$ are nonnegative.

The remainder of this paper is organized as follows. In Section 2.1, we primarily give the three regularization terms then the low-rank-based semisupervised subspace clustering model is proposed. In Section 2.2, we define an MCP- and SCAD-based norm to approximate rank function. Then a FCP-based nonconvex minimization model results for subspace clustering. In Section 3, for solving the proposed model, we present an algorithm that combines several approaches such as ADMM, LLA, weighted singular value thresholding, and so on. In Section 4, we conduct a series of simulations with several datasets to demonstrate the superiority of our method. In Section 5, we conclude this paper with some summation and future plans.

2. The Proposed Model

2.1. The Low-Rank Model Integrating Affinity and Clustering. We set $\mathbf{G}_i = [\mathbf{g}_1^i, \mathbf{g}_2^i, \ldots, \mathbf{g}_{n_i}^i] \in R^{c \times n_i}$ as the submatrix of $\mathbf{G}$ composed of the label vectors for all the data from subspace S_i. The ideal $\mathbf{G}_i$ is the elements in ith row are all one, and those in the other rows are all zero. With the expression $\mathbf{x}_j = \mathbf{X}\mathbf{z}_j\ (j = 1, 2, \ldots, n)$ and the ideal structure of $\mathbf{G}_i$, we can automatically seek the block diagonal structure of $\mathbf{Z}$ by minimizing the rank of matrix multiplication $\mathbf{G}\mathrm{Diag}(\mathbf{z}_j)$(refer to [31, 32] for the detailed interpretation). This sparsity between subspaces and density within a subspace implied by the block diagonal structure are preferred to the aim of subspace segmentation. So, we first give a minimization model as below. Here for every $j \in P\backslash S$, we set $I_j = \{i | g_{ij} \neq 0, i = 1, 2, \ldots, c\}$, and $\#I_j$ denotes the number of elements in I_j.

$$\begin{cases} \min\limits_{\mathbf{g}_j\,(j \in P\backslash S) \in \Delta_+, \mathbf{Z}} & \left\{ \dfrac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \gamma \sum\limits_{j \in P} \mathrm{rank}\left(\mathbf{G}\mathrm{Diag}\left(\mathbf{z}_j\right)\right) \right\} \\ \text{s.t.} & \text{For every } j \in P\backslash S, \#I_j = 1 \text{ and } I_j \text{ corresponds to the subspace the data } \mathbf{x}_j \text{ is from.} \end{cases} \tag{4}$$

It is rational for the label vector $\mathbf{g}_j$ to have only one nonzero element since, in real application; one data only belongs to one subspace. For example, it is impossible for one face image belonging to two persons. Since $\mathbf{g}_j$ has only one nonzero element if and only if $\|\mathbf{g}_j\|_1 - \|\mathbf{g}_j\|_2 = 0$. Under the constraint of $\mathbf{g}_j\,(j \in P\backslash S) \in \Delta_+$, its unique nonzero element has a value of one. For the simplicity of computing, we relax $\|\mathbf{g}_j\|_1 - \|\mathbf{g}_j\|_2 = 0$ as enabling the difference of l_1 norm and l_2 norm of $\mathbf{g}_j$ to achieve minimization and use as a regularized constraint. The problem becomes

$$\begin{cases} \min\limits_{\mathbf{g}_j\,(j \in P\backslash S) \in \Delta_+, \mathbf{Z}} & \left\{ \dfrac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \alpha \sum\limits_{j \in P\backslash S} \left(\left\|\mathbf{g}_j\right\|_1 - \left\|\mathbf{g}_j\right\|_2 \right) + \gamma \sum\limits_{j \in P} \mathrm{rank}\left(\mathbf{G}\mathrm{Diag}\left(\mathbf{z}_j\right)\right) \right\} \\ \text{s.t.} & \text{for every } j \in P\backslash S, I_j \text{ corresponds to the subspace the data } \mathbf{x}_j \text{ is from.} \end{cases} \tag{5}$$

Now the problem is how to make sure the nonzero element of $\mathbf{g}_j$ properly associate with the subspace the query data come. To do this, we use the following term:

$$\frac{\beta}{2}\sum_{i\in P\backslash S \text{ or } j\in P\backslash S}\left\|\mathbf{g}_i-\mathbf{g}_j\right\|_2^2\left(\left|Z_{ij}\right|+\left|Z_{ji}\right|\right)+\frac{1}{2}\sum_{i\in S}\sum_{j\in S}\left\|\mathbf{g}_i-\mathbf{g}_j\right\|_2^{2p}\left(\left|Z_{ij}\right|+\left|Z_{ji}\right|\right). \tag{6}$$

We denote elements of $\mathbf{Z}$ as $Z_{ij}\,(i,j\in\{1,2,\ldots,n\})$. The quantity of $|Z_{ij}|+|Z_{ji}|$ describes the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. We analyze from the following several aspects:

(1) For the case $i\in P\backslash S$, $j\in S$, or $i\in S$, $j\in P\backslash S$, that is, one of $\mathbf{x}_i$ and $\mathbf{x}_j$ already has a label. Taking $i\in P\backslash S$, $j\in S$ for instance, since $\mathbf{x}_j$ already has a label, $\mathbf{x}_i$ should belong to the same subspace with $\mathbf{x}_j$ as $|Z_{ij}|+|Z_{ji}|$ is large enough. This can be achieved by the first part of formula (6). In this case, the index of the nonzero elements of the unknown label $\mathbf{g}_i$ can be properly correlated to the subspace to which the data belongs.

(2) As neither $\mathbf{x}_i$ and $\mathbf{x}_j$ has a label, the first part of (6) also works for this case. Since both $\mathbf{g}_i$ and $\mathbf{g}_j$ have only one nonzero element, indexes of their nonzero elements must be the same as $|Z_{ij}|+|Z_{ji}|$ is large enough. Therefore, data $\mathbf{x}_i$ and $\mathbf{x}_j$ can be clustered into the same subspace, but which subspace cannot be decided. Figure 1 shows labels can be properly propagated from labeled data to unlabeled data; then the index of the nonzero element of each label can correspond to the accurate subspace even in this case.

(3) If both $\mathbf{x}_i$ and $\mathbf{x}_j$ have labels, which means $\mathbf{g}_i$ and $\mathbf{g}_j$ do not need to be solved, we only do not know the representation coefficient. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same subspace, namely, $\|\mathbf{g}_i-\mathbf{g}_j\|_2^2=0$, $|Z_{ij}|$ and $|Z_{ji}|$ should be large. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are not in the same subspace, i.e., $\|\mathbf{g}_i-\mathbf{g}_j\|_2^2=2$, $|Z_{ij}|$ and $|Z_{ji}|$ had better be zero. To this end, we use the second part of (6). Here the parameter p needs to be taken a large value.

Using all the data $\mathbf{x}_j\,(j=1,2,\ldots,n)$ as nodes, $\mathbf{x}_i$ and $\mathbf{x}_j$ have a connection between them as $|Z_{ij}|+|Z_{ji}|$ is large enough, and no connection as $|Z_{ij}|+|Z_{ji}|$ is small; then we obtain a graph of all data. Since for each data, the most similar data must come from the same subspace with it, each node must have a connection with at least one node in the same subspace. When there is at least one labeled data in each subspace, the label can be propagated from the labeled node to unlabeled nodes through formula (6). Therefore, the connected nodes can share the same label. This can be simply illustrated by Figure 1, in which there are only two classes and two labeled data (one for each subspace).

So far, we obtain the complete low-rank-based semi-supervised subspace clustering model

$$\min_{\mathbf{g}_j\,(j\in P\backslash S)\in\Delta_+\mathbf{Z}}\left\{\begin{array}{l}\dfrac{1}{2}\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2+\alpha\displaystyle\sum_{j\in P\backslash S}\left(\left\|\mathbf{g}_j\right\|_1-\left\|\mathbf{g}_j\right\|_2\right)+\left\{\dfrac{\beta}{2}\displaystyle\sum_{\substack{i\in PS\text{ or}\\ j\in P\backslash S}}\left\|\mathbf{g}_i-\mathbf{g}_j\right\|_2^2\left(\left|Z_{ij}\right|+\left|Z_{ji}\right|\right)\right.\\ \left.+\dfrac{1}{2}\displaystyle\sum_{i\in S}\sum_{j\in S}\left\|\mathbf{g}_i-\mathbf{g}_j\right\|_2^{2p}\left(\left|Z_{ij}\right|+\left|Z_{ji}\right|\right)\right\}+\gamma\displaystyle\sum_{j\in P}\operatorname{rank}\left(\mathbf{G}\operatorname{Diag}\left(\mathbf{z}_j\right)\right)\end{array}\right\}. \tag{7}$$

The minimization problem (7) is difficult to solve due to the combined nature of rank function. The standard approach is to replace rank function with the nuclear norm. Considering the fact that matrix nuclear norm is prone to overpenalize large singular values and thus usually leads to a biased estimation and the advantages of FCP over l_1 norm described in lots of works, we will utilize the FCP of singular value vector of a matrix to approximate rank function; thus, the model (7) is transformed into our model (3) presented in the introduction section, where matrix norm $\|\cdot\|_{P_{\lambda,a}}$ is defined as follows:

$$\|\mathbf{I}\|_{P_{\lambda,a}}=\sum_{k=1}^{c}P_{\lambda,a}\left(\sigma_k\left(\mathbf{I}\right)\right). \tag{8}$$

$\sigma_k(\mathbf{I})$ is the kth singular value of matrix $\mathbf{I}\in R^{c\times n}$ with $c\le n$. We choose the function $P_{\lambda,a}(\cdot)$ as $P_{\lambda,a}^{\mathrm{MCP}}(\cdot)$ or $P_{\lambda,a}^{\mathrm{SCAD}}(\cdot)$ defined as following separately. MCP is of the following form:

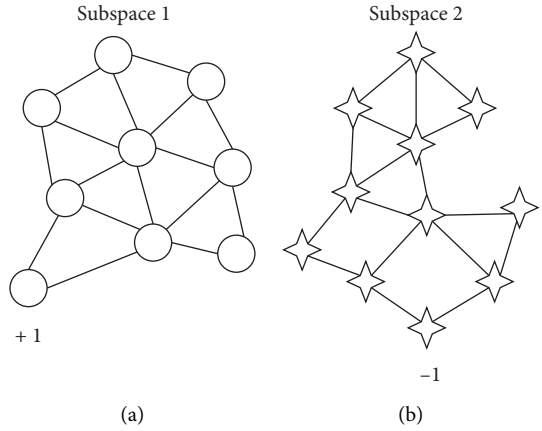

Figure 1: Semisupervised subspace clustering.

$$P_{\lambda,a}^{\mathrm{MCP}}(t) \coloneqq \begin{cases} \dfrac{a\lambda^2}{2} & \text{if } |t| \geq a\lambda \\ \lambda|t| - \dfrac{t^2}{2a} & \text{otherwise} \end{cases}, \quad (a > 1). \tag{9}$$

SCAD is of the following form:

$$P_{\lambda,a}^{\mathrm{SCAD}}(t) \coloneqq \begin{cases} \lambda|t| & |t| \leq \lambda \\ \dfrac{a\lambda|t| - 0.5\left(|t|^2 + \lambda^2\right)}{a-1} & \lambda < |t| < a\lambda \\ \dfrac{\lambda^2(a+1)}{2} & |t| \geq a\lambda \end{cases}, \quad (a > 2). \tag{10}$$

The matrix FCP norm defined in (8) satisfies the following properties.

Proposition 1. *For $a \in (1, \infty)$(MCP) or $a \in (2, \infty)$(SCAD), then*

(1) $\|\mathbf{I}\|_{P_{\lambda,a}} \geq 0$, *with equality iff* $\mathbf{I} = \mathbf{0}$

(2) $\|\mathbf{I}\|_{P_{\lambda,a}} \leq \lambda\|\mathbf{I}\|_*$, *and* $\lim_{a \longrightarrow \infty} \|\mathbf{I}\|_{P_{\lambda,a}} = \lambda\|\mathbf{I}\|_*$

(3) $\|\mathbf{I}\|_{P_{\lambda,a}}$ *is unitarily invariant, that is,* $\|\mathbf{UIV}\|_{P_{\lambda,a}} = \|\mathbf{I}\|_{P_{\lambda,a}}$ *whenever* $\mathbf{U}$ *and* $\mathbf{V}$ *are orthonormal*

(4) $\|\mathbf{I}\|_{P_{\lambda,a}}$ *is concave w. r. t. matrix* $|\mathbf{I}|$, *where* $|\mathbf{I}| = [|I_{ij}|]$

Property (1) is obvious. The second property can be easily verified since both SCAD and MCP penalty functions are upper bounded by Lasso penalty function and tend to Lasso penalty function as $a \longrightarrow \infty$. The third property is true due to the fact that singular values are not changed from $\mathbf{I}$ to $\mathbf{UIV}$ whenever $\mathbf{U}$ and $\mathbf{V}$ are orthonormal. The fourth property holds because of the concavity of the FCP penalty function. It is worth noting that the matrix FCP is not a real norm since it does not satisfy the triangle inequality of a norm.

3. Optimization

To solve problem (3), the original problem is converted to the following equivalent problem:

$$\min_{\substack{\mathbf{g}_j\,(j\in P\setminus S)\in\Delta_+,\\ \mathbf{Z},\mathbf{Q},\mathbf{y}_j\,(j\in P\setminus S),\\ \mathbf{J}_j\,(j\in P)}} \left\{ \begin{array}{c} \frac{1}{2}\|\mathbf{X}-\mathbf{XZ}\|_F^2+\alpha\sum\limits_{j\in P\setminus S}\left(\|\mathbf{y}_j\|_1-\|\mathbf{g}_j\|_2\right)+\gamma\sum\limits_{j\in P}\|\mathbf{J}_j\|_{P_{\lambda,a}} \\ +\frac{\beta}{2}\sum\limits_{i\in P\setminus S \text{ or } j\in P\setminus S}\|\mathbf{g}_i-\mathbf{g}_j\|_2^2\left(|Q_{ij}|+|Q_{ji}|\right) \\ +\frac{1}{2}\sum\limits_{i\in S}\sum\limits_{j\in S}\|\mathbf{g}_i-\mathbf{g}_j\|_2^{2p}\left(|Q_{ij}|+|Q_{ji}|\right) \end{array} \right\} \tag{11}$$

$$\text{s.t.}\begin{cases} \mathbf{J}_j=\mathbf{G}\text{Diag}\left(\mathbf{z}_j\right), & j\in P, \\ \mathbf{Q}=\mathbf{Z}, & \\ \mathbf{y}_j=\mathbf{g}_j, & j\in P\setminus S. \end{cases}$$

The solution to problem (11) is difficult to achieve directly. We adopt the Augmented Lagrangian Multiplier (ALM) scheme to derive the following unconstrained optimization problem:

$$\min_{\substack{\mathbf{g}_j\,(j\in P\setminus S)\in\Delta_+,\\ \mathbf{Z},\mathbf{Q},\mathbf{y}_j\,(j\in P\setminus S),\\ \mathbf{J}_j\,(j\in P)}} \left\{ \begin{array}{c} \frac{1}{2}\|\mathbf{X}-\mathbf{XZ}\|_F^2+\alpha\sum\limits_{j\in P\setminus S}\left(\|\mathbf{y}_j\|_1-\|\mathbf{g}_j\|_2\right) \\ +\frac{\beta}{2}\sum\limits_{i\in P\setminus S \text{ or } j\in P\setminus S}\|\mathbf{g}_i-\mathbf{g}_j\|_2^2\left(|Q_{ij}|+|Q_{ji}|\right) \\ +\frac{1}{2}\sum\limits_{i\in S}\sum\limits_{j\in S}\|\mathbf{g}_i-\mathbf{g}_j\|_2^{2p}\left(|Q_{ij}|+|Q_{ji}|\right)+\gamma\sum\limits_{j\in P}\|\mathbf{J}_j\|_{P_{\lambda,a}} \\ +\sum\limits_{j\in P}\text{tr}\left(\mathbf{Y}_j^T\left(\mathbf{G}\text{Diag}\left(\mathbf{z}_j\right)-\mathbf{J}_j\right)\right)+\text{tr}\left(\mathbf{F}^T\left(\mathbf{Z}-\mathbf{Q}\right)\right) \\ +\sum\limits_{j\in P\setminus S}\mathbf{u}_j^T\left(\mathbf{y}_j-\mathbf{g}_j\right)+\frac{\mu}{2}\sum\limits_{j\in P}\|\mathbf{G}\text{Diag}\left(\mathbf{z}_j\right)-\mathbf{J}_j\|_F^2 \\ +\frac{\mu}{2}\|\mathbf{Z}-\mathbf{Q}\|_F^2+\frac{\mu}{2}\sum\limits_{j\in P\setminus S}\|\mathbf{y}_j-\mathbf{g}_j\|_2^2 \end{array} \right\}, \tag{12}$$

where $\mathbf{Y}_j\in R^{c\times n}\,(j\in P)$, $\mathbf{F}\in R^{n\times n}$, $\mathbf{u}_j\in R^c\,(j\in P\setminus S)$ are Lagrangian multipliers and $\mu>0$ is the penalty parameter. Instead of optimizing all arguments simultaneously, as $\mathbf{g}_j\,(j\in P\setminus S)$, $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{y}_j\,(j\in P\setminus S)$, $\mathbf{J}_j\,(j\in P)$ are separable, we solve them individually and iteratively for $k=0,1,2,\ldots$.

(1) By fixing $\mathbf{g}_j^k\,(j \in P\backslash S)$, $\mathbf{Q}^k$, $\mathbf{y}_j^k\,(j \in P\backslash S)$, $\mathbf{J}_j^k\,(j \in P)$, we optimize every column $\mathbf{z}_j\,(j \in P)$ of $\mathbf{Z}$ by the following subproblem:

$$\mathbf{z}_j^{k+1} = \arg\min_{\mathbf{z}_j} \left\{ \begin{array}{l} \frac{1}{2}\left\|\mathbf{x}_j - \mathbf{X}\mathbf{z}_j\right\|_2^2 + \mathrm{tr}\left(\left(\mathbf{Y}_j^k\right)^T \mathbf{G}^k \mathrm{Diag}\left(\mathbf{z}_j\right)\right) + \left(\mathbf{f}_j^k\right)^T \mathbf{z}_j \\ \\ + \frac{\mu^k}{2}\left(\left\|\mathbf{G}^k \mathrm{Diag}\left(\mathbf{z}_j\right) - \mathbf{J}_j^k\right\|_{\mathrm{F}}^2 + \left\|\mathbf{z}_j - \mathbf{q}_j^k\right\|_2^2\right) \end{array} \right\}, \tag{13}$$

here $\mathbf{f}_j^k$ is jth column of $\mathbf{F}^k$ and $\mathbf{q}_j^k$ is jth column of $\mathbf{Q}^k$. Then the solution can be achieved via solving a linear system as follows:

$$\begin{aligned} &\left(\mathbf{X}^T\mathbf{X} + \mu^k \mathrm{Diag}\left(\left(\mathbf{G}^k\right)^T \mathbf{G}^k\right) + \mu^k \mathrm{I}\right)\mathbf{z}_j^{k+1} \\ &\quad = \mathbf{X}^T\mathbf{x}_j + \mathrm{diag}\left(\left(\mu^k\left(\mathbf{J}_j^k\right)^T - \left(\mathbf{Y}_j^k\right)^T\right)\mathbf{G}^k\right) + \mu^k\mathbf{q}_j^k - \mathbf{f}_j^k. \end{aligned} \tag{14}$$

(2) By fixing $\mathbf{Z}^{k+1}$, $\mathbf{Q}^k$, $\mathbf{y}_j^k\,(j \in P\backslash S)$, $\mathbf{J}_j^k\,(j \in P)$, we optimize every $\mathbf{g}_j\,(j \in P\backslash S)$ by the following subproblem:

$$\mathbf{g}_j^{k+1} = \arg\min_{\mathbf{g}_j\,(j\in P\backslash S)\in\Delta_+} \left\{ \begin{array}{l} -\alpha \sum\limits_{j\in P\backslash S} \left\|\mathbf{g}_j\right\|_2 + \frac{\beta}{2} \sum\limits_{i\in P\backslash S \text{ or } j\in P\backslash S} \left\|\mathbf{g}_i - \mathbf{g}_j\right\|_2^2 \left(\left|Q_{ij}^k\right| + \left|Q_{ji}^k\right|\right) \\ \\ + \sum\limits_{i\in P} \mathrm{tr}\left(\left(\mathbf{Y}_i^k\right)^T \mathbf{G}\mathrm{Diag}\left(\mathbf{z}_i^{k+1}\right)\right) - \sum\limits_{j\in P\backslash S} \left(\mathbf{u}_j^k\right)^T \mathbf{g}_j + \\ \\ \frac{\mu^k}{2}\left(\sum\limits_{i\in P} \left\|\mathbf{G}\mathrm{Diag}\left(\mathbf{z}_i^{k+1}\right) - \mathbf{J}_i^k\right\|_F^2 + \sum\limits_{j\in P\backslash S} \left\|\mathbf{y}_j^k - \mathbf{g}_j\right\|_2^2\right) \end{array} \right\}. \tag{15}$$

The above problem can be solved using the following equation:

$$\mathbf{g}_j^{k+1} = \frac{2\beta\sum_{i\in P}\left(\left|Q_{ji}^k\right| + \left|Q_{ij}^k\right|\right)\mathbf{g}_i^k + \sum_{i\in P}\left(\mu^k\mathbf{J}_{i,j}^k - \mathbf{Y}_{i,j}^k\right)Z_{ji}^{k+1} + \mathbf{u}_j^k + \mu^k\mathbf{y}_j^k}{2\beta\sum_{i\in P}\left(\left|Q_{ji}^k\right| + \left|Q_{ij}^k\right|\right) + \mu^k\sum_{i\in P}\left(Z_{ji}^{k+1}\right)^2 + \mu^k - \alpha\left(1/\left\|\mathbf{g}_j^k\right\|_2\right)}. \tag{16}$$

Here, $\mathbf{J}_{i,j}^k$ and $\mathbf{Y}_{i,j}^k$ represent jth column of matrix $\mathbf{J}_i^k$ and $\mathbf{Y}_i^k$ separately. Z_{ji}^{k+1} is the jth element of vector $\mathbf{z}_i$. $\mathbf{g}_j^{k+1}$ is then projected onto the set $\Delta_+ = \{\mathbf{g}|\mathbf{g} = (g_k)_{k=1}^c, g_k \geq 0, \sum_{k=1}^c g_k = 1\}$ by the algorithm presented by [33].

(3) For $i \in P\backslash S$ or $j \in P\backslash S$, the optimized Q_{ij}^{k+1} can be obtained using a weighted soft-thresholding algorithm

$$\frac{\beta}{2\mu^k}\left\|\mathbf{g}_i^{k+1} - \mathbf{g}_j^{k+1}\right\|_2^2\left|Q_{ij}\right| + \frac{1}{2}\left(Q_{ij} - \left(Z_{ij}^{k+1} + \frac{F_{ij}^k}{\mu^k}\right)\right)^2. \tag{17}$$

As $i \in S$ and $j \in S$, since $\mathbf{g}_i$ and $\mathbf{g}_j$ are known, we get Q_{ij} from the following equation:

$$\frac{1}{2\mu^k}\left\|\mathbf{g}_i - \mathbf{g}_j\right\|_2^{2p}\left|Q_{ij}\right| + \frac{1}{2}\left(Q_{ij} - \left(Z_{ij}^{k+1} + \frac{F_{ij}^k}{\mu^k}\right)\right)^2. \tag{18}$$

(4) To update $\mathbf{y}_j\,(j \in P\backslash S)$, the following subproblem is solved:

$$\mathbf{y}_j^{k+1} = \arg\min_{\mathbf{y}_j} \frac{\alpha}{\mu^k}\left\|\mathbf{y}_j\right\|_1 + \frac{1}{2}\left\|\mathbf{y}_j - \left(\mathbf{g}_j^{k+1} - \frac{\mathbf{u}_j^k}{\mu^k}\right)\right\|_2^2. \tag{19}$$

Problem (18) can be solved by the soft-thresholding operator.

(5) To update $\mathbf{J}_j\,(j \in P)$, the following subproblem is solved:

$$\mathbf{J}_j^{k+1} = \arg\min_{\mathbf{J}_j} \frac{\gamma}{\mu^k}\left\|\mathbf{J}_j\right\|_{P_{\lambda,a}} + \frac{1}{2}\left\|\mathbf{J}_j - \left(\mathbf{G}^{k+1}\mathrm{Diag}\left(\mathbf{z}_j^{k+1}\right) + \frac{\mathbf{Y}_j^k}{\mu^k}\right)\right\|_F^2, \tag{20}$$

which can be solved through LLA by using the result of singular value thresholding as the initial value for $\mathbf{J}_j$.

The Lagrangian multipliers are updated as follows:

$$\mathbf{Y}_j^{k+1} = \mathbf{Y}_j^k + \mu^k\left(\mathbf{G}^{k+1}\mathrm{Diag}\left(\mathbf{z}_j^{k+1}\right) - \mathbf{J}_j^{k+1}\right), \quad j \in P, \tag{21}$$

$$\mathbf{F}^{k+1} = \mathbf{F}^k + \mu^k\left(\mathbf{Z}^{k+1} - \mathbf{Q}^{k+1}\right), \tag{22}$$

$$\mathbf{u}_j^{k+1} = \mathbf{u}_j^k + \mu^k\left(\mathbf{y}_j^{k+1} - \mathbf{g}_j^{k+1}\right), \quad j \in P\backslash S. \tag{23}$$

Steps (14)–(23) are repeated until the convergence conditions are attained. In this algorithm, the penalty parameter μ_k starts with $\mu^0 = 10^{-9}$, then $\mu^{k+1} = \max(\rho\mu^k, 10^9)$ in each iteration step with $\rho = 1.1$.

4. Experiments

To evaluate the performance of our proposed method, we conduct experiments on five benchmark datasets: the face image dataset Extended Yale B [34] and ORL [35], the object image dataset COIL20 [36], the handwriting number image dataset MNIST [37], and handwriting number and letter image dataset Alphadigits. We directly cite some results reported in [12] on the dataset ORL and Alphadigits. We compare our proposed method with ten baseline algorithms, including SSC, kernel sparse subspace clustering (KSSC) [38], SSC by orthogonal matching pursuit (SSC-OMP) [39], S^3C [40], SSRSC [41], LRR, low-rank subspace clustering (LRSC) [42], efficient dense subspace clustering (EDSC) [43], TLRR [44], and KTRR [12]. Among these methods, SSC, KSSC, SSC-OMP, S^3C, and SSRSC are sparse-based and LRR, LRSC, EDSC, TLRR are low-rank-based; KTRR is based on ridge regression.

4.1. Datasets. For the datasets used in our experiments, we briefly describe these datasets as follows: (1) Extended Yale B. The Extended Yale B Database consists of 2432 face images in total from 38 subjects, with 64 frontal face images per subject taken under different illumination conditions. Face images of each subject are a low-dimensional subspace. In our experiments, each image is downsampled from 192×168 to 32×32 pixels. (2) ORL contains face images of size 32×32 pixels from 40 individuals. Each individual has 10 images taken at different times, with varying facial expressions, facial details, and lighting conditions. (3) COIL20 contains 1440 grayscale images of 20 objects. Each image was downsampled to 32×32. (4) MNIST contains 70,000 images of handwritten digits 0–9 with a size of 28×28 pixels. (5) Alphadigits dataset is a binary dataset, which collects handwritten digits 0–9 and letters A–Z. Totally, there are 36 classes and 39 samples for each class, of which each example has a size of 20×16 pixels.

4.2. Evaluation Metrics. We use three evaluation metrics to testify the effectiveness of the proposed method, including clustering accuracy (CA), normalized mutual information (NMI), and purity. All experiments run ten times using n random choices of subjects for each time. For each subject, we randomly select t images, of which m images are prelabeled in our method. Then the average CA, NMI, and purity are reported. We compute the clustering accuracy as follows:

$$\text{clustering accuracy} = \frac{\text{number of correctly clustered data}}{n(t-m)}. \tag{24}$$

Clustering accuracy measures the accuracy rate of clustering. NMI measures the quality of the clusters. Purity measures the extent to which each cluster contains samples from primarily the same subject. Higher values of these metrics indicate better clustering quality. More details of the last two metrics can be found in [45]. The best results are shown in bold font. Experiments verify that there is no obvious difference between MCP and SCAD when they are applied to our model, so we do not list them separately. In all of our experiments, we use the intensity feature of each image and stretch it to a column of the data matrix $\mathbf{X}$.

4.3. Parameter Selection. The parameters for all the methods are adjusted to obtain the best clustering results. Our method involves six parameters a, λ, α, β, γ, and p. Among these parameters, a and λ are associated with the FCP penalty functions. In our experiments, $a = 5$ and $\lambda = 1$ can obtain the best results. α, β, and γ are used to balance the roles of three regularization terms. The best choice for α is 12.5735 ± 0.0115, for β is 100 ± 0.8, and for γ is $7.005 \times 10^{-3} \pm 2.5 \times 10^{-5}$. Parameter p needs to be set large enough to penalize the large representation coefficients when two labeled data have different labels. We fix $p = 20$ which can achieve the best results for all the following experimental settings.

4.4. Initialization and Stopping Criterion. All the variables $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{g}_j\,(j \in P\backslash S)$, $\mathbf{y}_j\,(j \in P\backslash S)$ and Lagrange multipliers $\mathbf{Y}_j\,(j \in P)$, $\mathbf{F}$, $\mathbf{u}_j\,(j \in P\backslash S)$ are started with all elements being zero. $\mathbf{J}_j\,(j \in P)$ is initialized to $\mathbf{G}^0\mathrm{Diag}(\mathbf{z}_j^0)$ where $\mathbf{G}^0$ consists of the known labels and initial unknown labels $\mathbf{g}_j^0\,(j \in P\backslash S)$. We use three errors $\sum_{j\in P}\|\mathbf{G}\mathrm{Diag}(\mathbf{z}_j) - \mathbf{J}_j\|_F$, $\|\mathbf{Z} - \mathbf{Q}\|_F$, $\sum_{j\in P\backslash S}\|\mathbf{y}_j - \mathbf{g}_j\|_2$ all less than 10^{-6} as the stopping criterion of iterations.

4.5. Performances. The average CA, NMI, and purity of all the methods on the two face image datasets; i.e., Extended Yale B and ORL, are reported in Tables 1 and 2 separately. For the database Extended Yale B, we use $n \in \{5, 10\}$ random choices of subjects for each time. For each subject, we

Table 1: Performance on Extended Yale B.

Methods	SSC	LRR	LRSC	KSSC	SSC-OMP	EDSC	S^3C	SSRSC	TLRR	KTRR	Our method	
						CA (%)						
$n = 5$												
$t = 10$	79.3	77.8	71.2	78.2	79.2	81.3	80.5	77.4	71.3	81.8	$m = 2$	**82.5**
$t = 20$	82.7	80.4	74.8	80.8	81.2	83.6	82.9	78.6	73.2	83.8	$m = 4$	**84.2**
$t = 30$	85.7	84.7	78.2	84.5	85.9	86.3	85.4	81.3	76.8	87.0	$m = 5$	**87.8**
$n = 10$												
$t = 10$	76.9	66.5	61.2	72.8	74.6	79.0	78.5	75.0	69.7	79.5	$m = 2$	**80.2**
$t = 20$	78.3	67.9	63.3	75.3	75.4	81.8	80.6	76.1	71.3	81.4	$m = 4$	**82.3**
$t = 30$	82.4	70.3	65.1	77.4	80.7	84.7	73.4	79.6	74.3	84.5	$m = 6$	**84.9**
						NMI						
$n = 5$												
$t = 10$	0.3998	0.5351	0.4195	0.3749	0.4034	0.5845	0.5290	0.4671	0.4023	0.6018	$m = 2$	0.6122
$t = 20$	0.4272	0.5543	0.4433	0.4034	0.4213	0.6229	0.5439	0.4734	0.4194	0.6326	$m = 4$	0.6433
$t = 30$	0.4483	0.5834	0.4584	0.4323	0.4549	0.6434	0.5745	0.5032	0.4483	0.6693	$m = 5$	0.6853
$n = 10$												
$t = 10$	0.4837	0.5511	0.5533	0.4783	0.4855	0.5584	0.5531	0.5321	0.4199	0.6242	$m = 2$	0.6329
$t = 20$	0.5048	0.5738	0.5754	0.4893	0.5034	0.5745	0.5633	0.5583	0.4284	0.6383	$m = 4$	0.6508
$t = 30$	0.5438	0.6182	0.6137	0.5032	0.5138	0.6056	0.5844	0.5822	0.4409	0.6594	$m = 6$	0.6683
						Purity						
$n = 5$												
$t = 10$	0.5491	0.6582	0.5667	0.5286	0.5543	0.6683	0.6382	0.6182	0.5469	0.6970	$m = 2$	0.7094
$t = 20$	0.5438	0.6643	0.5985	0.5454	0.5567	0.6782	0.6433	0.6283	0.5589	0.7093	$m = 4$	0.7233
$t = 30$	0.5582	0.6802	0.6087	0.5563	0.5758	0.6963	0.6646	0.6504	0.5761	0.7276	$m = 5$	0.7586
$n = 10$												
$t = 10$	0.5191	0.5818	0.4693	0.5084	0.5208	0.6087	0.5582	0.5591	0.4413	0.6536	$m = 2$	0.6676
$t = 20$	0.5263	0.6062	0.4876	0.5294	0.5389	0.6128	0.5723	0.5590	0.4592	0.6685	$m = 4$	0.6698
$t = 30$	0.5376	0.6265	0.5083	0.5509	0.5518	0.6385	0.5868	0.5634	0.4703	0.6856	$m = 6$	0.6865

Table 2: Performance on ORL.

Methods	SSC	LRR	LRSC	KSSC	SSC-OMP	EDSC	S^3C	SSRSC	TLRR	KTRR	Our method	
						CA (%)						
$n = 10$												
$t = 10$	77.6	74.9	68.5	76.2	78.4	82.9	81.2	78.2	76.8	87.5	$m = 2$	**88.3**
$n = 20$												
$t = 10$	78.5	73.8	67.3	75.6	77.5	81.6	81.6	79.1	75.4	89.0	$m = 2$	**89.3**
$n = 30$												
$t = 10$	79.1	73.8	65.4	73.0	74.7	79.1	82.8	79.9	74.7	90.4	$m = 2$	**90.8**
						NMI						
$n = 10$												
$t = 10$	0.7711	0.8376	0.7239	0.7434	0.7676	0.8239	0.8410	0.8118	0.7965	0.8428	$m = 2$	**0.8656**
$n = 20$												
$t = 10$	0.8271	0.8197	0.7094	0.7832	0.8156	0.8053	0.8408	0.8313	0.7889	0.8804	$m = 2$	**0.9056**
$n = 30$												
$t = 10$	0.8355	0.8372	0.7323	0.8184	0.8382	0.8338	0.8663	0.8359	0.7726	0.9038	$m = 2$	**0.9210**
						Purity						
$n = 10$												
$t = 10$	0.7520	0.7860	0.7364	0.7455	0.7564	0.7993	0.8100	0.8020	0.7913	0.8330	$m = 2$	**0.8450**
$n = 20$												
$t = 10$	0.7610	0.7620	0.7049	0.7385	0.75636	0.7564	0.7785	0.7700	0.7311	0.8355	$m = 2$	**0.8444**
$n = 30$												
$t = 10$	0.7483	0.7640	0.6687	0.7448	0.7411	0.7563	0.7937	0.7610	0.6773	0.8500	$m = 2$	**0.8665**

randomly select t images, where $t \in \{10, 20, 30\}$, among which m images are prelabeled in our method. For ORL, we randomly choose $n \in \{10, 20, 30\}$ persons. Each individual has 10 images. With only a small portion of data being prelabeled, our method can achieve comparative results with the most state-of-the-art methods.

The previous experiment targets face clustering. To show the generality of our algorithm, we also evaluate it on the COIL20 object image dataset. The results are reported in

Table 3: Performance on COIL20.

Methods	SSC	LRR	LRSC	KSSC	SSC-OMP	EDSC	S^3C	SSRSC	TLRR	KTRR	Our method	
						CA (%)						
$n = 10$												
$t = 10$	76.9	70.8	70.6	70.9	61.6	76.6	75.4	73.2	71.8	76.9	$m = 2$	**77.3**
$t = 20$	77.6	66.0	65.4	71.4	62.5	78.0	77.8	74.6	71.7	78.5	$m = 4$	**79.4**
$t = 30$	80.4	70.2	70.4	74.9	64.5	80.1	79.5	76.3	72.1	80.4	$m = 5$	**81.6**
						NMI						
$n = 10$												
$t = 10$	0.5223	0.4778	0.4743	0.4832	0.4244	0.5203	0.5137	0.4893	0.4622	0.5365	$m = 2$	**0.5483**
$t = 20$	0.5332	0.4223	0.4532	0.5087	0.4238	0.5591	0.5329	0.5128	0.4723	0.5732	$m = 4$	**0.5832**
$t = 30$	0.5538	0.4765	04808	0.5248	0.4391	0.5624	0.5593	0.5328	0.4933	0.6030	$m = 5$	**0.6037**
						Purity						
$n = 10$												
$t = 10$	0.4639	0.4334	0.4103	0.4283	0.4083	0.4729	0.4855	0.4734	0.4492	0.4932	$m = 2$	**0.5039**
$t = 20$	0.4896	0.4034	0.4093	0.4768	0.4186	0.4833	0.4916	0.4845	0.4692	0.5243	$m = 4$	**0.5302**
$t = 30$	0.5007	0.4340	0.4237	0.4876	0.4297	0.5029	0.5184	0.5045	0.4704	0.5508	$m = 5$	**0.5638**

Table 4: Performance on MNIST.

Methods	SSC	LRR	LRSC	KSSC	SSC-OMP	EDSC	S^3C	SSRSC	TLRR	KTRR	Our method	
						CA (%)						
$n = 10$												
$t = 30$	78.05	64.7	63.1	74.4	73.5	78.7	60.6	79.6	68.4	80.8	$m = 5$	**81.8**
$t = 40$	84.4	67.0	65.3	79.8	79.6	84.8	62.7	81.4	71.7	84.9	$m = 7$	**85.9**
$t = 50$	86.8	70.3	68.0	82.8	84.8	88.4	65.8	83.2	74.5	88.8	$m = 8$	**90.3**
						NMI						
$n = 10$												
$t = 30$	0.7284	0.6148	0.6094	0.7034	0.7056	0.7493	0.4333	0.7302	0.6183	0.7500	$m = 5$	**0.7508**
$t = 40$	0.7603	0.6394	0.6339	0.7544	0.7306	0.7684	0.4354	0.7495	0.6443	0.7745	$m = 7$	**0.7865**
$t = 50$	0.7734	0.6549	0.6407	0.7596	0.7385	0.7693	0.4426	0.7528	0.6487	0.7946	$m = 8$	**0.7905**
						Purity						
$n = 10$												
$t = 30$	0.7339	0.6594	0.6283	0.7084	0.6948	0.7648	0.5245	0.7257	0.6384	0.7747	$m = 5$	**0.7806**
$t = 40$	0.7644	0.6645	0.6366	0.7585	0.7058	0.7743	0.5433	0.7504	0.6748	0.7947	$m = 7$	**0.8019**
$t = 50$	0.7796	0.6765	0.6444	0.7658	0.7399	0.8085	0.5574	0.7653	0.6737	0.8095	$m = 8$	**0.8145**

Table 5: Performance on Alphadigits.

Methods	SSC	LRR	LRSC	KSSC	SSC-OMP	EDSC	S^3C	SSRSC	TLRR	KTRR	Our method	
						CA (%)						
$n = 5$												
$t = 39$	80.7	79.5	77.3	80.5	81.2	81.3	60.8	82.6	70.4	84.1	$m = 6$	**84.7**
$n = 10$												
$t = 39$	79.5	77.4	75.1	78.4	79.5	79.2	58.4	79.5	69.8	81.8	$m = 6$	**82.6**
						NMI						
$n = 5$												
$t = 39$	0.6519	0.6567	0.5842	0.6432	0.6578	0.6684	0.2810	0.6467	0.4980	0.6959	$m = 6$	**0.7121**
$n = 10$												
$t = 39$	0.5525	0.6521	0.5546	0.6194	0.6323	0.6328	0.2497	0.5864	0.4981	0.6650	$m = 6$	**0.6790**
						Purity						
$n = 5$												
$t = 39$	0.6715	0.6974	0.7054	0.6682	0.6738	0.7036	0.3841	0.6638	0.5619	0.7277	$m = 6$	**0.7329**
$n = 10$												
$t = 39$	0.5056	0.6158	0.6476	0.6396	0.6543	0.6782	0.2504	0.5258	0.4551	0.6305	$m = 6$	**0.6406**

Table 3. In contrast with the previous human face datasets, in which faces are well aligned and have similar structures, the object images from COIL20 are more diverse, and even samples from the same object differ from each other due to the change of viewing angle. This makes this dataset challenging for subspace clustering techniques. Experimental results are listed in Table 3. For each of the chosen 10 subjects, we randomly select $t \in \{10, 20, 30\}$ images, of which

$m \in \{2, 4, 5\}$; images already have labels for our method. Similar to the above datasets, the result of our method is the best among all the eleven methods.

To further verify the effectiveness of our proposed method, we conduct experiments on two datasets containing handwritten digits, i.e., MNIST and Alphadigits. The Alphadigits database also contains 26 letters. For the MNIST database, we randomly choose t images for each digit 0–9, where $t \in \{30, 40, 50\}$, and then apply all the methods to cluster the images. For our methods, $m \in \{5, 7, 8\}$ images are prelabeled of the t images for each digit. For the Alphadigits database, we use $n \in \{5, 10\}$ random choices of subjects. For each subject, we use all the images, among which $m = 6$ images are prelabeled for our method. The CA, NMI, and purity are reported in Tables 4 and 5. Our method gets the best clustering result among all the compared methods.

5. Conclusion

We propose a novel nonconvex formulation for the subspace segmentation problem. In our work, the labels of all data are directly solved from the model rather than using spectral clustering algorithms. We give two regularization constraints about the label vectors. One is to force the label vector only to have one nonzero element by minimizing the difference of l_1 norm and l_2 norm. Another is to make sure data from the same subspace have the same label as much as possible so that the index of that nonzero element of each label vector can indicate the subspace from which the data come. Due to many advantages of FCP over l_1 norm, we present an FCP-based low-rank approximation norm and apply it to the combination of semisupervised label matrix and representation vector. This term can enforce the representation matrix better meeting block diagonal structure. The labels and the representation matrix are contained in two regularization terms. Therefore, they can interact and promote each other during the computing process. We give a solving algorithm based on ADMM, LLA, weighted singular value thresholding, weighted soft thresholding, and so on.

In the future, we will continue conducting research on estimation or learning methods for the parameters in our proposed model because we have to tune those parameters in order to achieve better results. It is possibly better to integrate the classical knowledge-based approaches into the deep learning architecture, making the algorithm enjoy both the flexibility of the deep learning-based methods and the clear structure of the classical approaches.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61772389), Natural Science Basic Research Program of Shaanxi Province, China (2020 JM-569, 2021 JM-440, and 2020 JQ-817), Science and Technology Plan of Shaanxi Province (2020 GY-066), Principal Fund Program of Xi'an Technological University (XAGDXJJ17027), Science and Technology Plan of Weiyang District, Xi'an, Shaanxi Province (201925), and Key Research and Development Program of Shaanxi (2021 GY-137).

References

[1] Q. Zhang and C. Peng, "Feature selection embedded robust K-means," *IEEE Access*, vol. 8, pp. 166164–166175, 2020.

[2] C. Peng, Z. Kang, and S. Cai, "Integrate and conquer: double-sided two-dimensional k-means via integrating of projection and manifold construction," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, 2018.

[3] A. Y. Ng, M. I. Jordan, Y. Weiss et al., "On spectral clustering: analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849–856, 2002.

[4] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[5] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2931–2940, IEEE, Honolulu, Hawaii, July 2017.

[6] M. Yin, Yi Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5157–5164, Las Vegas, NV, USA, July 2016.

[7] J. Liu, Y. Chen, J. Zhang, and Z. Xu, "Enhancing low-rank subspace clustering by manifold regularization," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 4022–4030, 2014.

[8] V. M. Patel, H. Van Nguyen, and R. Vidal, "Latent space sparse and low-rank subspace clustering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 691–701, 2015.

[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[10] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[11] P. Ji, T. Zhang, H. Li et al., "Deep subspace clustering networks," in *Proceedings of the NIPS'17*, Long Beach, CA, USA, December 2017.

[12] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proceedings of the IJCAI*, New York, NY, USA, July 2016.

[13] C. Peng, Q. Zhang, K. Zhao, C. Chen, and Q. Cheng, "Kernel two-dimensional ridge regression for subspace clustering," *Pattern Recognition*, vol. 113, no. 3, Article ID 107749, 2020.

[14] C. Peng and Q. Cheng, "Discriminative ridge machine: a classifier for high-dimensional data or imbalanced data," *IEEE Transactions on Neural Networks and Learing Systems*, https://arxiv.org/abs/1904.07496v2, 2020.

[15] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, 2018.

[16] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[17] E. J. Candes, X. Li, Yi Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–11, 2009.

[18] E. Candes and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, 2010.

[19] C. Peng, K. Zhao, H. Li, and Q. Cheng, "Subspace clustering using log-determinant rank approximation, KDD 2015 KDD '15," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 925–934, Sydney, Australia, August 2015.

[20] C. Peng, Y. Chen, Z. Kang, C. Chen, and Q. Cheng, "Robust principal component analysis: a factorization-based approach with linear complexity," *Information Sciences*, vol. 513, pp. 581–599, 2020.

[21] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its Oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[22] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$L1 = 2$ regularization: a thresholding representation theory and a fast solver," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[23] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[24] C. H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, pp. 894–942, 2010.

[25] P. Breheny and J. Huang, "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection," *The Annals of Applied Statistics*, vol. 5, no. 1, pp. 232–253, 2011.

[26] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A nonconvex relaxation approach to sparse dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, June 2011.

[27] Z. Zhang and B. Tu, "Nonconvex penalization using laplace exponents and concave conjugates," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Denver, CO, USA, December 2012.

[28] Z. Zhang, S. Wang, D. Liu, and M. I. Jordan, "EP-GIG priors and applications in Bayesian sparse learning," *Journal of Machine Learning Research*, vol. 13, pp. 2031–2061, 2012.

[29] C. Zhang and T. Zhang, "A general theory of concave regularization for high dimensional sparse estimation problems," *Statistical Science*, vol. 27, no. 4, pp. 576–593, 2012.

[30] J. Fan, L. Xue, and H. Zou, "Strong oracle optimality of folded concave penalized estimation," *Annals of Statistics*, vol. 42, no. 3, pp. 819–849, 2014.

[31] J. Lai and X. Jiang, "Supervised trace lasso for robust face recognition," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1–6, IEEE, Chengdu, China, July 2014.

[32] Z. Wenjuan, F. Xiangchu, and C. Yunmei, "A manifold Laplacian regularized semi-supervised sparse image classification method with a variant trace lasso norm," *IEEE Access*, vol. 8, pp. 97361–97369, 2020.

[33] Y. Chen and X. Ye, "Projection onto a simplex," *Mathematics*, https://arxiv.org/abs/1101.6081, 2011.

[34] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[35] F. S. Samaria and A. C. H.. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Sarasota, FL, USA, December 1994.

[36] S. A. Nene, S. K. Nayar, and H. Murase, *Columbia Object Image Library*, Columbia University, New York, NY, USA, 1996.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proceedings of IEEE International Conference on Image Processing*, pp. 2849–2853, Paris, France, January 2014.

[39] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3918–3927, Las Vegas, NV, USA, July 2016.

[40] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: a unified optimization framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 277–286, Boston, MA, USA, June 2015.

[41] J. Xu, M. Yu, L. Shao et al., "Scaled simplex representation for subspace clustering," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1493–1505, 2021.

[42] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.

[43] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering," in *Proceedings of the IEEE Winter Conference on Applications of Computerision*, pp. 461–468, Steamboat Springs, CO, USA, March 2014.

[44] P. Zhou, C. Lu, J. Feng, Z. Lin, and S. Yan, "Tensor low-rank representation for data recovery and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, 2019.

[45] C. Peng, Z. Zhang, Z. Kang et al., "Nonnegative matrix factorization with local similarity learning," *Information Sciences*, vol. 562, 2021.