Subspace Clustering by Integrating Sparseness and Spatial-Closeness Priors

Zhe Li¹, Haodong Pei²,³*, Liang He¹, Jiaming Liu¹, Jiaxin Hu¹ and Dongji Wang⁴,⁵

¹Futian Power Supply Bureau, Shenzhen Power Supply Bureau Co., Ltd, Shenzhen, Guangdong 518001, China
²Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences, Shanghai 200083, China
³Shanghai Institute of Technical Physics of the Chinese Academy of Sciences, Shanghai 200083, China
⁴National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
⁵School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
Email: peihaodong@sina.com

Abstract. How to construct an effective sample affinity matrix is an important problem for subspace clustering, and most existing subspace clustering algorithms pursue the affinity matrix in a single space. In this paper, we propose a novel computational framework for subspace clustering, called Complementary Subspace Clustering (CSC) at first, where the affinity matrix is constructed in a pair of complementary spaces which provide different and complementary constraints on the affinity matrix. Many existing structural priors on self representation and dimensionality reduction can be seamlessly integrated into the CSC framework. Then under this framework, we explore a simple and effective subspace clustering algorithm by respectively introducing two basic priors - sparse representation and spatial closeness - into the referred pair of spaces. Moreover, a kernel variant of the proposed clustering algorithm is present. Extensive experimental results demonstrate that although only basic priors are involved, the explored algorithms from the CSC framework can improve the clustering performances significantly when the number of the sample classes is relatively big.

1. Introduction
In recent years, subspace clustering has attracted much attention in machine learning and computer vision, and numerous subspace clustering algorithms have been proposed in the literature. Existing algorithms can be roughly grouped into four categories [1]: algebraic algorithms, iterative algorithms, statistical algorithms, and spectral-clustering-based algorithms. In this paper, we focus on the spectral-clustering-based algorithms [1-11], where the main challenge is to design an effective affinity matrix measuring the similarity between pairwise samples. In the following, we mainly review some works on this topic.

An early work on spectral clustering [2] constructed the affinity matrix according to the computed singular vectors via the Singular Value Decomposition of a feature track matrix. The Local Subspace Affinity (LSA) algorithm [12] was proposed based on the assumption that each sample and its $k$ nearest neighbours generally lied in the same subspace, and it constructed the affinity of pairwise samples based
on the principal angles between their estimated local subspaces. The Locally Linear Manifold Clustering (LLMC) algorithm [13] was explored by fitting a local subspace to each sample and its $k$ nearest neighbours, where the computed coefficients were used to construct the affinity matrix. One key problem on both LSA and LLMC is that it is hard for them to assign an appropriate number $k$ of neighbours.

More recently, the self-representation idea, assuming that each sample in the input sample set can be represented as a linear combination of the other samples, has been extensively adopted for improving algorithmic performances [1, 4-11]. One prominent advantage of these self-representation-based algorithms is that the ‘neighbours’ of each sample can be identified without assigning a special neighbour number $k$. Most of these algorithms consists of two steps: Firstly, the representation coefficient matrix is computed by solving a derived optimization problem from some assumptions. Secondly, the affinity matrix is directly constructed according to the computed coefficient matrix $Z$, and spectral clustering algorithms, e.g. [14-15], are employed on the affinity matrix to cluster the input data into different groups.

The main differences among these self-representation algorithms are the introduced regularizers on the representation coefficient matrix from different assumptions. Elhamifar and Vidal [1] proposed the Sparse Subspace Clustering (SSC) algorithm with an $l_1$-norm regularizer on the coefficient matrix, based on the assumption that a sample can be represented as a linear combination of a small number of the rest samples. Liu et. al. [8] proposed the Low-Rank Representation (LRR) algorithm for subspace clustering, where a nuclear-norm regularizer was involved. Luo et al. [6] proposed the Multiple Subspace Representation (MSR) algorithm, which employed both the sparseness prior and the low-rank prior. Lu et al. [7] proposed two Least Squares Regression (LSR) algorithms with a Frobenius-norm regularizer. Hu et al. [11] proposed the Smooth Representation (SMR) clustering algorithm with a smooth regularizer for enforcing the grouping effect. In addition, Patel et al. [16] proposed an algorithm for simultaneous dimensionality reduction and data clustering, which computed a representation coefficient matrix in a low-dimensional latent space.

It is noted from the above discussions that there are two common factors which probably weaken the performances of these existing works: (1) Most existing works pursue an affinity matrix in a single data space, e.g. the original space [1, 6, 8], a latent low-dimensional space [16]. As is known, there is no information loss when samples are handled in the original space, but since it is believed that the high-dimensional samples generally lie near a low-dimensional manifold, these high-dimensional samples usually contain much redundant information which may weaken the clustering performances. Conversely, when samples are handled in a low-dimensional space, clean and compact representations can be obtained, but some discriminant information of these samples may be lost after dimensionality reduction, since it is hard to select an appropriate dimension for achieving a tradeoff between compact representation and information loss. (2) Most existing works pursue an affinity matrix by imposing a single assumption on the affinity matrix, such as sparse representation [1], low-rank representation [8], etc. But a single assumption often cannot reflect the intrinsic structure of the input samples sufficiently, especially in the cases of insufficient samples and noisy samples.

Motivated by the above two factors, in this paper, we propose a novel subspace clustering framework for pursuing a more effective affinity matrix, called complementary subspace clustering (CSC), where the affinity matrix is computed with different and complementary priors on the affinity matrix (or intermediate variables dependent on it) in both the original space and a latent space. Under the CSC framework, we currently focus on investigating further potentials of existing structural priors rather than exploring a new prior, hence, we propose a novel clustering algorithm based on the two motivated basic assumptions - sparse representation and spatial closeness (as described in Section 2.2), called SRSC. Then we give a detailed theoretical analysis on the characteristics of SRSC, which explains why this algorithm with only two basic assumptions can achieve effective clustering in the cases of insufficient data and noisy data. Furthermore, a kernel variant of the SRSC algorithm incorporating nonlinearity is explored, called KSRSC. In summary, there are two main contributions in this work:
• The CSC framework for subspace clustering, integrating different and complementary constraints in two different spaces, is able to unify many existing works (as discussed in Section 2.1). Moreover, it provides a general platform to explore new subspace clustering algorithms, and many existing structural priors on both data representation and dimensionality reduction can be seamlessly integrated in this framework.

• The SRSC algorithm and its kernel variant KSRSC are proposed under the explored CSC framework. A theoretical analysis is provided to support the effectiveness of the proposed algorithms for clustering. Our experimental results show that although only two basic structural priors are involved in the proposed algorithms, their clustering accuracies are significantly enhanced when the number of the sample classes is relatively big.

2. Complementary Subspace Clustering

In this section, the complementary subspace clustering framework (CSC) is described at first. Then under the CSC framework, the SRSC algorithm is proposed by calculating the affinity matrix in both the original space and a latent space, and followed by a detailed theoretical analysis. Finally, the KSRSC algorithm, the kernel variant of the SRSC algorithm, is explored.

2.1. Framework Description

The CSC framework consists of two steps - affinity matrix construction and spectral clustering. Unlike most existing works that build affinity matrices in a single space, the key idea of the CSC framework is to construct an affinity matrix in both the original space and a latent space that provide different and complementary constraints on the affinity matrix (or intermediate variables dependent on it). Then, the classic spectral clustering algorithm Normalized Cuts [14] is utilized to ultimately cluster the input data with the obtained affinity matrix. The flowchart of the proposed CSC framework is shown in figure 1.

![Figure 1](image)

Figure 1. The flowchart of the proposed CSC framework for subspace clustering. Optional: PCA is optionally applied as a pre-processing step for computational convenience.

Let $X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{D \times N}$ be a set of $N$ samples drawn from $M$ linear subspace $S_1, S_2, \cdots, S_M \subset \mathbb{R}^D$. Let $\Gamma : \mathbb{R}^D \rightarrow \mathbb{R}^d$ be an unknown mapping from the input $D$-dimensional space to a $d$-dimensional latent space. Let $f(W, Z, \lambda) \perp X$ be an objective function in the input space, with
respect to the affinity matrix $W$ and a set of $n$ intermediate variables $Z_{i_1^n}$ satisfying $W = Y_i Z_{i_1^n}$, $i = 1, 2, \ldots, n$, $Y_i$ indicates a pre-defined function describing the relation between $W$ and $Z_{i_1^n}$, and $g \Gamma X , W ; Z_{i_1^n} | X$ be an objective function in a $d$-dimensional latent space with respect to $W$, $Z_{i_1^n}$ and $\Gamma$. Then the CSC framework can be formulated as:

$$
\min_{W, Z_{i_1^n}, \Gamma} \Phi W, Z_{i_1^n}, \Gamma = 1 - \eta f W, Z_{i_1^n} | X + \eta g \Gamma X , W , Z_{i_1^n} | X
$$

s.t. $\Gamma \in C_r, Z_{i_1} \in C_Z, W = Y_i Z_{i_1}, i = 1, 2, \ldots, n$

where $C_r$ indicates the constraint set on $\Gamma$, $C_Z$ indicates the constraint set on $Z_{i_1^n}$, $\eta \in 0, 1$ is a tuning constant.

As seen from equation (1), $f$ is to impose a regularizer on the affinity matrix $W$ and intermediate variables $Z_{i_1^n}$ in the original space, while $g$ is to impose a regularizer on $W$ and $Z_{i_1^n}$ in a latent space. And $\eta$ sets the trade-off between these two regularizers. Therefore, the proposed framework is able to compute the affinity matrix by utilizing complementary constraints on the input data in two different spaces.

Relation to previous works: It is noted that despite the derived different objective functions from different motivations of many existing subspace clustering algorithms ([1, 5-8, 16], etc.), equation (1) is able to unify them within a common framework.

Table 1 lists the corresponding configurations on $\eta, f, \Gamma, g, Y_i, Z, C_r, C_Z$ to formulate six existing self-representation based algorithms in the proposed CSC framework. It has to be pointed out that for these six algorithms, only the representation coefficient matrix $Z$ (i.e. $Z_{i_1^n}$ with $n=1$ in equation (1)), rather than the affinity matrix $W$, is used as the variable in their objective functions, hence, the configurations on $Y_i$ with $n=1$ for them are ineffective to solve their objective functions.

### Table 1. Configurations on $\{\eta, f, \Gamma, g, Y_i, Z, C_r, C_Z\}$ in the CSC framework for formulating six existing self-representation based methods.

| $\eta$ | $f$ | $\Gamma$ | $g$ | $Y_i$ | $C_r, C_Z$ |
|--------|-----|---------|-----|------|------------|
| SSC    | 0   | $\|Z\|$ | -   | $W = \|Z\| + \|Z^\top\|$ | $Z | X = XZ, diag Z = 0$ |
| LRR    | 0   | $\|Z\|$ | -   | $W = \|Z\| + \|Z^\top\|$ | $Z | X = XZ$ |
| MSR    | (0, 1) | $\|Z\|$ | Identity mapping | $\|Z\|$ | $Z | X = XZ, diag Z = 0$ |
| LSR1   | 0   | $\|Z\|^2_F$ | -   | $W = \|Z\| + \|Z^\top\| / 2$ | $Z | X = XZ, diag Z = 0$ |
| LSR2   | 0   | $\|Z\|^2_F$ | -   | $W = \|Z\| + \|Z^\top\| / 2$ | $Z | X = XZ$ |
| LS3C   | 1   | $P \in \mathbb{R}^{d \times d}$ | - | $P \in \mathbb{R}^{d \times d}$ | $P | P^T P = I$ |

2.2. SRSC Algorithm

As described above, the CSC framework provides a general platform to explore new algorithms for subspace clustering, and many existing structural priors on self representation and dimensionality reduction in the literature can be seamlessly integrated into the proposed framework, such sparse
representation, low-rank representation, etc. Hence, we currently investigate further potentials of these existing priors for subspace clustering under the CSC framework, and we propose a new subspace clustering algorithm by employing two very basic priors, called SRSC.

(1) Problem equation: The proposed algorithm is based on the following two assumption:

Assumption 1: Sparse Representation (SR): In the original space, each sample can be represented in a linear combination of a small number of intraclass samples which are selected automatically via the principle of sparsity.

Assumption 2: Spatial Closeness (SC): In a latent low-dimensional space, the projected points, corresponding to any pair of ‘adjacent’ points in the original space, are spatially close.

The SR assumption is directly from the popular sparse coding theory, while the SC assumption is motivated by the observation from the dimensionality reduction field that two adjacent intraclass points in a high-dimensional space are prone to be spatially close in some low-dimensional space.

The SR assumption is characterized by the term $h$:

$$ h = \|Z\| \quad s.t. \quad X = XZ, \quad diag(Z) = 0 $$

where $\|Z\| = \sum_{i,j} |Z_{ij}|$, $Z$ is the coefficient matrix.

Let $W = |Z| + |Z^T| / 2$ be the affinity matrix. Under the SC assumption, a projection matrix $P \in \mathbb{R}^{D \times d}$ is pursued so that any pair of projected points with a large affinity is prone to be spatially close. Hence, the SC assumption is characterized by the following weighted-sum term $g$:

$$ g = \sum_{i,j} W_{ij} \|P^T x_i - P^T x_j\|_F $$

$$ s.t. \quad W = \frac{1}{2} |Z| + |Z^T|, \quad P^TP = I $$

In order to ensure that both equations (2) and (3) are available simultaneously in the CSC framework, the parameter $\eta$ in equation (1) is required to belong to $0, 1$, then by adding equations (2) and (3) into equation (1), we have the following equation:

$$ \min_{W, Z, P} \Phi = \|Z\| + \zeta \sum_{i,j} W_{ij} \|P^T x_i - P^T x_j\|_F^2 $$

$$ s.t. \quad X = XZ, diag(Z) = 0, W = \frac{1}{2} |Z| + |Z^T|, P^TP = I $$

where $\zeta = \eta / 1 - \eta > 0$.

Considering the cases of noisy samples and insufficient samples in real applications, we reformulate equation (4) as:

$$ \min_{W, Z, P} \Phi' = \|X - XZ\|_F + \lambda \|Z\| + \zeta \sum_{i,j} W_{ij} \|P^T x_i - P^T x_j\|_F^2 $$

$$ s.t. \quad diag(Z) = 0, W = \frac{1}{2} |Z| + |Z^T|, P^TP = I $$

(2) Optimization: Here, an iterative procedure is performed for computing the projection matrix $P$, the coefficient matrix $Z$, and the affinity matrix $W$ by solving equation (5). At each iteration, each of these variables is updated by fixing the rests. The detailed performance of the iterative procedure is
described as follows:

Initializing $Z$ and $W$: Let $Z_0$ and $W_0$ be the initials of $Z$ and $W$ respectively, and $\tilde{Z}_0 \in \mathbb{R}^{(N-1) \times (N-1)}$ be the submatrix of $Z_0$ by removing the diagonal entries of $Z_0$. Let $\bar{z}_{0,i}$ be the $i$-th column of $\tilde{Z}_0$, and $\bar{X}_i = [x_i, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]$. Then $\bar{z}_{0,i}$ is simply computed by

$$\bar{z}_{0,i} = \bar{X}_i^\dagger x_i$$  \hspace{1cm} (6)

where $\dagger$ indicates the pseudo inverse. After $Z_0$ is obtained, $W_0 = \frac{1}{2}(Z_0 + Z_0^T)$.

Updating $P$: Fixing $Z$ and $W$ in equation (5), we have

$$\min_P \sum_{i,j} W_{ij} \left\| P^T x_i - P^T x_j \right\|_2^2, \hspace{0.5cm} s.t. \hspace{0.5cm} P^T P = I$$  \hspace{1cm} (7)

By simple algebra equation (7) can be rewritten as

$$\min_P \text{trace} \ P^T XLX^T P, \hspace{0.5cm} s.t. \hspace{0.5cm} P^T P = I$$  \hspace{1cm} (8)

where $L = D - W$ is the Laplacian matrix, and $D$ is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. Then, an optimal $P$ can be obtained by solving the following eigenvalue problem

$$XLX^T P = \mu P$$  \hspace{1cm} (9)

Updating $Z$ and $W$: For computational convenience, an auxiliary non-negative symmetric matrix $A \in \mathbb{R}^{N \times N}$ is introduced with $A_{ij} = \left\| P^T x_i - P^T x_j \right\|_2^2$. Fixing $P$ and replacing $W$ with $\frac{1}{2}(Z + Z^T)$ in equation (5), we have

$$\min_{W,Z,P} \left\| X - XZ \right\|_F^2 + \lambda \left\| Z \right\|_F^2 + \zeta \sum_{i,j} |Z_{ij}| A_{ij}$$

$$s.t. \hspace{0.5cm} \text{diag}(Z) = 0$$  \hspace{1cm} (10)

Then equation (10) can be reformulated as a standard $l_1$-norm optimization problem:

$$\min_{W,Z,P} \left\| X - XZ \right\|_F^2 + \sum_{i,j} \lambda + \zeta A_{ij} |Z_{ij}|$$

$$s.t. \hspace{0.5cm} \text{diag}(Z) = 0$$  \hspace{1cm} (11)

Equation (11) can be easily solved by applying many existing $l_1$-norm solvers [17, 18]. In our experiments, the solver [17] is employed for solving equation (11). Then, the affinity matrix $W$ is updated directly according to $W = \frac{1}{2}(Z + Z^T)$.

Here, the convergence criterion is $|\Phi W \cdot [Z, Z^T, P]^T - \Phi W, Z, P^T| < \varepsilon$ where $\varepsilon$ is a pre-set threshold (empirically, $\varepsilon = 10^{-5}$). Once the affinity matrix $W$ is obtained, the Normalized Cuts algorithm [14] is utilized to ultimately cluster the input data with $W$.

(3) Theoretical analysis: Let $B$ be a permutation matrix so that $\hat{X} = XB = [X_1, X_2, \ldots, X_M]$, where $X_i$ is a matrix whose columns are the input $N_i$ samples drawn from the $i$-th subspace $S_i$, and let $\hat{W} = B^{-1}WB$, $\hat{Z} = B^{-1}ZB$. Let $W^*, Z^*, P^*$ be an optimal solution to equation (5) (or (4)) with respect to $W, Z, P$, and let $\hat{W}^* = B^{-1}W^*B$, $\hat{Z}^* = B^{-1}Z^*B$. Then by simple algebra equation, it hold
that $\hat{W}^*, \hat{Z}^*, P^*$ is an optimal solution to equation (5) (or (4)) when $X$ is replaced with $\hat{X}$. Hence without loss of generality, we assume that the input matrix $X = X_1, X_2, \ldots, X_M$ is in the general position.

In the case of sufficient data sampling and noise-free data, we have the following proposition:

Proposition 1: If the subspaces are independent and the data sampling is sufficient, the optimal $Z^*$ to the problem (4) is block diagonal regardless of $P$:

$$Z^* = \begin{bmatrix}
Z_{b,11} & 0 & \cdots & 0 \\
0 & Z_{b,22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & Z_{b,MM}
\end{bmatrix}$$

(12)

where $Z_{b,ii}$ $i = 1, \ldots, M \in \mathbb{R}^{N_i \times N_i}$ corresponding to $X_i$, $N_i$ is the number of the samples drawn from the $i$-th subspace $S_i$.

In the case of insufficient data sampling, we have the following proposition:

Proposition 2: If the subspaces are orthogonal, the optimal $Z^*$ to the problem (5) is block diagonal regardless of $P$.

Proposition 1 shows that the obtained coefficient matrix by solving equation (4) has a block diagonal structure under the assumption that the subspaces are independent and the data sampling is sufficient. And proposition 2 shows that the obtained coefficient matrix by solving equation (5) has a block diagonal structure under the assumption that the subspaces are orthogonal.

However, the above referred assumptions are too strong to hold in many real applications where data are drawn from dependent subspaces and the data sampling is insufficient. Hence, we further analyse the characteristics of the proposed algorithm under a general condition, and give the following two propositions:

Proposition 3: If $Z_{ij}^*$ (the element in the $i$-th row and $j$-th column of $Z^*$) $\rightarrow \infty$, we have $\|P^T x_i - P^T x_j\|_F^2 \rightarrow 0$.

Proposition 4: For an arbitrary pair of points $x_i, x_j \in X$, if $\|P^T x_i - P^T x_j\|_F^2 \rightarrow \infty$, we have $Z_{ij}^* \rightarrow 0$.

In the following, we use these two propositions to analyse the characteristics of the proposed algorithm under a general condition regardless of drawn data from dependent spaces and insufficient data sampling.

Compatibility between the used two assumptions: It is noted that (1) The bigger $Z_{ij}^*$ is, the closer the corresponding low-dimensional projected points are prone to be as inferred from Proposition 3, which is in accord with the SC assumption, and simultaneously, the more likely the corresponding original points belong to a same class based on the SR assumption. (2) The more distant two low-dimensional projected points are, the smaller their corresponding coefficient $Z_{ij}^*$ is prone to be as inferred from Proposition 4, which is in accord with the SR assumption, and simultaneously, the more likely the corresponding original points belong to two different classes based on the SC assumption. Therefore, the SC and SR assumptions are compatible, and they can exist together successfully in the proposed algorithm.

Effects of $g$ and $h$ for complementary subspace clustering: As shown in equation (1)-(3), (5), complementary subspace clustering generally aims to construct an affinity matrix in a pair of spaces rather than a single space, where two complementary objective terms $g$ and $h$ can be employed.
simultaneously. The effect of \( g \) is that: at each iteration, given a computed affinity matrix \( W \) (also \( Z ) \) where a large affinity is more likely to indicate that the corresponding pair of points belongs to a same class based on Assumption 1, the updated projection matrix \( P \) by solving (9) is prone to encourage the intraclass points with large affinities closer in the latent space according to Proposition 3. The effect of \( h \) is that: at each iteration, given a computed projection matrix \( P \) by solving (9) that makes most projected intraclass points relatively close in the latent space, the weights (i.e., \( \lambda + \zeta A_{ij} \) in equation (11)) for the corresponding representation coefficients to the pairs of intraclass points are decreased accordingly. Naturally, the weights for the corresponding coefficients to the pairs of interclass points are relatively increased, consequently, these updated coefficients corresponding to the interclass point pairs by minimizing equation (11) are prone to be decreased according to Proposition 4. Therefore, both \( h \) and \( g \) are implemented for decreasing the corresponding coefficients to the interclass point pairs during the iterative procedure.

Convergence: The SRSC algorithm is designed with a standard alternating descent procedure involving an eigenvalue decomposition step and an \( l_1 \)-optimization step, hence, the solution to equation (5) is prone to converge to a local optimum dependent on the convergence of the used solver [17], whose convergence has been proved in Ref. [17]. And the experimental results in figure 2 also show that SRSC always converges to a local minimum in few iterations.

Advantage: One important problem for most existing self-representation-based algorithms is that in the cases of insufficient data sampling and drawn data from dependent subspaces, some non-block-diagonal elements (i.e., the coefficients corresponding to interclass point pairs) in the computed coefficient matrix by these algorithms may be relatively large, which may bring seriously negative influences on clustering. Compared with these algorithms, the main advantage of SRSC is that it is able to decrease the values of these non-block-diagonal elements regardless of insufficient data sampling and drawn data from dependent subspaces to some extent, due to the fact that at each iteration, the term \( g \) in equation (3) is encouraged to pay more attention to penalize the affinity of each pair of interclass points that are far away from each other by utilizing the above referred effects of \( h \) and \( g \), which is also demonstrated by our experimental results in figure 3.

2.3. Kernel Variant of SRSC

It is considered that the performances of linear algorithms may be limited in some cases, and a possible way to improve their performances is to employ the kernel trick to map the input data to a higher-dimensional space and then project them back to a low-dimensional space. In this section, we explore a kernel variant of the proposed SRSC algorithm by incorporating nonlinearity, called KSRSC.

Let \( \phi : \mathbb{R}^d \rightarrow F \) be a nonlinear mapping from the input space to some high-dimensional feature space \( F \), and \( K \) be a kernel gram matrix with \( K_{ij} = k(x_i, x_j) \) where \( k(x_i, x_j) = \phi(x_i)^T \phi(x_j) \) is the kernel function. Let \( \alpha = [\alpha_{i,j}] \in \mathbb{R}^{N \times d} \) be a coefficient matrix, and \( P : F \rightarrow \mathbb{R}^d \) be a projection matrix from the high-dimensional feature space \( F \) to a \( d \)-dimensional latent space. Each column vector \( P_j, j = 1, 2, \ldots, d \) of \( P \) can be represented as a combination of the mapped high-dimensional data, i.e. \( P_j = \sum_{i=1}^N \alpha_{i,j} \phi(x_i) \). Then, fixing the SR term (2), we kernelize the SC term (3), and construct the following objective function similar to equation (5):

\[
\min_{W,Z,\alpha} \Phi(W, Z, \alpha) = \|X - XZ\|^2_F + \lambda \|Z\|^2_F + \zeta \sum_{i,j} W_{ij} \|\alpha^T \phi(x_i)^T \phi(x_j) - \alpha^T \phi(x_i)^T \phi(x_j)\|^2_F
\]

s.t. \( \text{diag}(Z) = 0, W = \frac{1}{2} |Z| + |Z^T|, \alpha^T K \alpha = I \) (13)

Similar to the operations in the proposed SRSC algorithm, equation (13) can also be optimized in an
iterative manner as follows. Both KSRSC and SRSC are outlined in Algorithm 1 (table 2), and the initials \( Z_0, W_0 \) for \( Z, W \) are computed in the same manner as SRSC.

Updating \( \alpha \): Fixing \( W, Z \), (13) can be reformulated as:

\[
\min_{\alpha} \text{trace} \alpha^T KLK \alpha \quad \text{s.t.} \quad \alpha^T K \alpha = 1
\]

(14)

Then, an optimal \( \alpha \) can be obtained by solving the following generalized eigenvalue problem:

\[
KLK \alpha = \mu K \alpha
\]

(15)

Updating \( Z \) and \( W \): An auxiliary non-negative symmetric matrix \( A \in \mathbb{R}^{N \times N} \) is introduced with \( A_{ij} = \| \alpha^T \phi X^T \phi x_i - \alpha^T \phi X^T \phi x_j \|^2 \). Fixing \( P \) and replacing \( W \) with \( |Z| + |Z'| / 2 \) in equation (13), we have an objective function with the same form as equation (11), which can be also solved by the solver [17].

**Table 2.** Pseudocode of SRSC and KSRSC Algorithms.

| Algorithm 1: SRSC & KSRSC |
|---------------------------|
| **Input:** \( X, t = 0 \) |
| **Output:** \( Z_t, W_t \) |
| 1 | Compute \( Z_0 \) according to (6). \( W_0 = \frac{|Z_0| + |Z'_0|}{2} \) ; |
| 2 | while not converge do |
| 3 | Update \( P_t \) or \( \alpha_t \) by solving (9) or (15) with \( W_t \) ; |
| 4 | Compute \( A \) with \( P_t \) or \( \alpha_t \), and update \( Z_t \) by solving (11). \( W_t = \frac{|Z_t| + |Z'_t|}{2} \) ; |
| 5 | \( t = t + 1 \) ; |
| 6 | end |
| 7 | Cluster \( X \) into different groups via Normalized Cuts. |

3. Experimental Results

In our experiments, the proposed SRSC and KSRSC are implemented on a Core 4 PC. Experiments on two popular benchmark databases in the literature - the Extended Yale Face B database [19] and the Hopkins 155 motion database [20] - are performed in comparison to four self-representation-based algorithms, SSR [1], LRR [21], LSR1 and LSR2 [7], as well as a dimensionality reduction algorithm LPP [22] that has a similar form to the term (3) of SRSC. Subspace clustering error (also motion segmentation error for the Hopkins 155 database) is used to measure the performances of all these algorithms, and it is defined as

\[
\text{error} = \frac{\text{number of misclassified points}}{\text{total number of points}} \times 100\%
\]

(16)

3.1. Face Clustering

The Extended Yale Face B database is used for evaluating the performances of all these algorithms on the face clustering problem, which contains face images from 38 individuals and around 64 images under different illuminations per individual. The images are resized into \( 32 \times 32 \), and then they are vectorized. We use the first \( c = 2, 3, 5, 8, 10 \) classes for face clustering. For the \( c \)-classes clustering problem, these original 1024-dimensional vectors are projected onto a \( D \)-dimensional \( (D = c \times 6) \) subspace by PCA at first, and all these algorithms are performed on the obtained \( D \)-dimensional data. The parameters for all these algorithms are manually tuned, and we report their best results. The corresponding parameter settings are: SSC - \((\lambda = 3.0)\); LRR - \((\lambda = 1.1)\); LSR1 and
LSR2 – ($\lambda = 0.4$ when $c = 2, 5$, $\lambda = 4 \times 10^{-3}$ when $c = \{3, 8, 10\}$); SRSC – ($\lambda = 7.5 \times 10^{-2}, \zeta = 5$); KSRSC – ($\lambda = 7.5 \times 10^{-2}, \zeta = 5$). And a polynomial kernel $k(x, y) = (\langle x, y \rangle + a)^d$ with the parameter setting $(a = 0.01, b = 1)$ is used in KSRSC.

Influence of latent-space dimension $d$ on clustering: The dimension $d$ of the latent space is important for our CSC framework, however, as is known, it is generally hard to achieve automatic dimension selection for such a dimensionality reduction problem as equation (7). Here, we investigate the influence of $d$ on face clustering and implement the SRSC algorithm with $d = \{5D / 6, 4D / 6, 3D / 6, 2D / 6\} = \{5c, 4c, 3c, 2c\}$ on the Extended Yale Face B database, and Table 3 shows the corresponding results. As is seen, when the class number $c$ is relatively small ($c = \{2, 3, 5, 8\}$), the clustering errors by SRSC with varying $d$ are quite close. When $c = 10$, the clustering errors by SRSC with $d = \{3c, 2c\}$ are significantly higher than those by SRSC with $d = \{5c, 4c\}$, probably because too much sample discriminant information may be lost after these samples are projected into a latent space with an excessively small dimension. However, it is worth noting from tables 3-5 (tables 4-5 list the best results by all these referred algorithms) that compared with these referred existing algorithms, SCSR with $d$ varying from $5D / 6$ to $2D / 6$ performs better in most cases, which demonstrates the SRSC algorithm is not quite sensitive to the latent-space dimension $d$. Therefore, in our rest experiments, $d$ is always set to $4D / 6$ for both SRSC and KSRSC.

**Table 3.** Clustering error (%) of SRSC with varying $d$ on the $c = \{2, 3, 5, 8, 10\}$ classes of data from extended Yale Face B.

| $c$ | $d = 5c$ | $d = 4c$ | $d = 3c$ | $d = 2c$ |
|-----|---------|---------|---------|---------|
| 2   | 0.78    | 0.78    | 0.78    | 0       |
| 3   | 1.04    | 0.52    | 1.56    | 1.56    |
| 5   | 1.56    | 1.88    | 2.19    | 2.19    |
| 8   | 1.56    | 1.17    | 3.32    | 3.91    |
| 10  | 2.81    | 1.56    | 11.88   | 16.41   |

**Figure 2.** Variation curves of $\Phi, W, Z, P$ with $d = \{5c, 4c, 3c, 2c\}$ on the $c = \{2, 3, 5, 8, 10\}$ classes of data from Extended Yale Face B.

In addition, in order to demonstrate the SRSC’s convergence with varying $d$, we plot the corresponding variation curves of the objective function (5) with $d = 5c, 4c, 3c, 2c$ on the $c = \{2, 3, 5, 8, 10\}$ classes of data in figure 2, which shows SRSC achieves convergence after 10 iterations or so. In fact, SRSC is always convergent in our experiments all the time.

Comparative evaluation: It is noted from Table 1 that for improving the clustering performances, an additional post processing step to scale the obtained coefficient matrix is implemented in SSC, LSR1, LSR2, and an additional post-processing step to compute the skinny SVD on the obtained coefficient matrix is implemented in LRR. However, the proposed SRSC and KSRSC do not employ any additional step. Hence, for comparing the performances of all these algorithms more clearly, we test SSC, LRR, LSR1, LSR2 with/without additional post-processing steps respectively. Table 4 lists the errors of LPP,
SRSC, KSRSC, as well as the other four algorithms without post-processing steps, and Table 5 lists the errors of LPP, SRSC, KSRSC, as well as the other algorithms with post-processing steps. Figure 3 shows their computed coefficient matrices on the 10-classes clustering problem.

![Figure 3. Computed coefficient matrices on the 10-classes clustering problem.](image)

**Table 4.** Clustering errors (%) of all these referred algorithms without additional post-processing steps on the $c = \{2,3,5,8,10\}$ classes of data from extended Yale Face B.

| $c$ | SSC | LRR | LSR1 | LSR2 | LPP | SRSC | KSRSC |
|-----|-----|-----|------|------|-----|------|-------|
| 2   | 1.56| 3.91| 3.91 | 3.91 | 0.78| 0.78 | 0     |
| 3   | 5.73| 6.25| 6.25 | 6.25 | 22.92| 0.52 | 0.52  |
| 5   | 10.31| 19.37| 18.12| 17.81| 16.25| 1.88 | 3.75  |
| 8   | 25.20| 24.61| 24.41| 20.51| 17.77| 1.17 | 4.88  |
| 10  | 22.50| 35.47| 33.12| 30.31| 17.03| 1.56 | 2.81  |

**Table 5.** Clustering errors (%) of the referred algorithms (SSC, LRR, LSR1, LSR2) with their own post-processing steps as well as LPP, SRSC, KSRSC on extended Yale Face B.

| $c$ | SSC | LRR | LSR1 | LSR2 | LPP | SRSC | KSRSC |
|-----|-----|-----|------|------|-----|------|-------|
| 2   | 1.56| 0.78| 3.12 | 3.12 | 0.78| 0.78 | 0     |
| 3   | 5.21| 0.52| 6.25 | 6.25 | 22.92| 0.52 | 0.52  |
| 5   | 7.81| 1.88| 12.81| 8.44 | 16.25| 1.88 | 3.75  |
| 8   | 15.43| 7.42| 22.07| 21.09| 17.77| 1.17 | 4.88  |
| 10  | 24.37| 19.84| 29.53| 26.09| 17.03| 1.56 | 2.81  |

As seen from Table 4, although SRSC and KSRSC are designed by employing two very basic structural priors to the explored CSC framework, these two algorithms outperform other referred algorithms without additional post-processing steps all the time. And as seen from Figure 3, there are less non-zero elements lying outside the diagonal blocks in the obtained coefficient matrices by both SRSC and KSRSC than those by other referred algorithms. These demonstrate that the proposed CSC framework is able to pursue more effective coefficient matrix by utilizing existing priors even if no novel priors are involved. Moreover, as seen from Table 5, when additional post-processing steps are implemented in SSC, LRR, LSR1, and LSR2, their performances are indeed improved, however, our algorithms still outperform these algorithms in most cases. With the increase of the class number, the count of the used samples is prone to be more insufficient for clustering, and our algorithms have a clearer advantage, which further demonstrates our analysis on the proposed algorithms in Section 2. It is particularly noted that compared with both SSC and LPP which have similar forms to the terms (2) and (3) of SRSC respectively, SRSC outperforms them with a large margin, probably due to its novel construction manner for the affinity matrix within a pair of spaces rather than a single space.

### 3.2. Motion Segmentation

Motion segmentation refers to clustering a video sequence into multiple spatiotemporal regions corresponding to different moving objects. Since a set of point trajectories are usually extracted from the input video in advance for further motion segmentation, the motion segmentation problem can be handled by performing subspace clustering algorithms on these extracted point coordinates. Here, the
Hopkins 155 database is used for evaluating the performances of all the referred algorithms on the motion segmentation problem. This database contains 156 video sequences of two or three motions with extracted feature points. The original data is projected onto a $D$-dimensional ($D = 12$) subspace by PCA as done in Ref. [7] at first, and then all these algorithms are performed on these $D$-dimensional data.

Here, in order to prevent other factors from interfering the fairness of the comparison, all the referred algorithms are implemented on the Hopkins 155 database without additional post-processing steps, and table 6 reports the max value (Max), mean value (Mean), median value (Med.), and the standard deviation (Std) of the computed segmentation errors by these algorithms. And the corresponding parameter settings for all these algorithms are: SSC $- (\lambda = 50)$; LRR $- (\lambda = 3.0)$; LSR1 $- (\lambda = 4.8 \times 10^{-3})$; LSR2 $- (\lambda = 4.6 \times 10^{-3})$; LPP $- (k = 15, \delta = 0.2)$; SRSC $- (\lambda = 7.5 \times 10^{-3}, \zeta = 0.5)$; KSRSC $- (\lambda = 3.75 \times 10^{-3}, \zeta = 2 \times 10^{-5})$. And a polynomial kernel with $(a = 0.01, b = 1)$ is used in KSRSC. As is seen, the computed mean errors by SRSC and KSRSC are lower than those by the other algorithms. SRSC has the lowest ‘Max’, ‘Mean’, ‘Med.’, and ‘Std’ of the segmentation errors. This demonstrates that compared with the other five algorithms, the computed coefficient matrices by the proposed algorithms are more effective for the motion segmentation problem.

**Table 6. Segmentation errors (%) on Hopkins 155 database.**

|       | SSC  | LRR  | LSR1 | LSR2 | LPP  | SRSC | KSRSC |
|-------|------|------|------|------|------|------|-------|
| Max   | 47.62| 41.15| 36.36| 36.36| 38.04| 30.08| 36.25 |
| Mean  | 3.27 | 4.36 | 2.50 | 2.84 | 4.38 | 2.05 | 2.48  |
| Med.  | 0    | 0.58 | 0.31 | 0.34 | 0    | 0    | 0     |
| Std   | 8.89 | 9.00 | 5.62 | 6.16 | 7.85 | 5.56 | 6.38  |

4. Conclusion

In this paper, a novel computational framework CSC is explored for subspace clustering, which constructs the affinity matrix in both the original data space and a latent space. The CSC framework is able to unify many existing works and provide a general platform to explore new clustering algorithms. Under the CSC framework, the SRSC algorithm and its kernel variant KSRSC are proposed based on two basic structural priors. Moreover, we provide a detailed theoretical analysis on the characteristics of the proposed algorithms, and the experimental results demonstrate the effectiveness of the proposed algorithms in comparison to five state-of-the-art algorithms.

Acknowledgments

This work was supported by the technology project of Shenzhen Power Supply Bureau Co., Ltd (No.0909002019030103FTPW00064), and the Open Research Fund from Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences.

References

[1] Elhamifar E and Vidal R 2013 Sparse subspace clustering: Algorithm, theory, and applications *IEEE Transaction on Pattern Analysis and Machine Intelligence* 35 (11) 2765-2781.
[2] Boult T and Brown L 1991 Factorization-based segmentation of motions *Proceedings of the IEEE Workshop on Visual Motion* 179-186.
[3] Vidal R, Tron R and Hartley R 2008 Multiframe motion segmentation with missing data using power factorization and gpca *International Journal of Computer Vision* 79 (1) 85-105.
[4] Wang S, Yuan X, Yao T, Yan S and Shen J 2011 Efficient subspace segmentation via quadratic programming *Proceedings of the National Conference on Artificial Intelligence* pp 519-524.
[5] Elhamifar E and Vidal R 2011 Sparse manifold clustering and embedding *Proceedings of 25th Annual Conference on Neural Information Processing Systems* pp 55-63.
[6] Luo D, Nie F, Ding C and Huang H 2011 Multi-subspace representation and discovery *European Conference ECML PKDD* pp 405-420.

[7] Lu C, Min H, Zhao Z, Zhu L, Huang D and Yan S 2012 Robust and efficient subspace segmentation via least squares regression *European Conference on Computer Vision* 347-360.

[8] Liu G, Lin Z, Yan S, Sun J, Yu Y and Ma Y 2013 Robust recovery of subspace structures by low-rank representation *IEEE Transaction on Pattern Analysis and Machine Intelligence* 35 (1) 171-184.

[9] Zhang Y, Sun Z, He R and Tan T 2013 Robust subspace clustering via half-quadratic minimization *Proceedings of International Conference on Computer Vision* pp 3096-3103.

[10] Feng J, Lin Z, Xu H and Yan S 2014 Robust subspace segmentation with block-diagonal prior *IEEE Conference on Computer Vision and Pattern Recognition* pp 23-37.

[11] Hu H, Lin Z, Feng J and Zhou J 2014 Smooth representation clustering *IEEE Conference on Computer Vision and Pattern Recognition* pp 1-8.

[12] Yan J and Pollefeys M 2006 A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate *European Conference on Computer Vision* pp 94-106.

[13] Goh A and Vidal R 2007 Segmenting motions of different types by unsupervised manifold clustering *IEEE Conference on Computer Vision and Pattern Recognition* pp 1-6.

[14] Shi J and Malik J 2000 Normalized cuts and image segmentation *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) 888-905.

[15] Ng A, Jordan M and Weiss Y 2001 On spectral clustering: Analysis and an algorithm *Conference and Workshop on Neural Information Processing Systems* pp 849-856.

[16] Patel V, Nguyen H and Vidal R 2013 Latent space sparse subspace clustering *IEEE International Conference on Computer Vision* pp 225-232.

[17] Kim S, Koh K, Lustig M, Boyd S and Gorinevsky D 2007 A method for large-scale $l_1$-regularized least squares *IEEE Journal on Selected Topics in Signal Processing* 1 (4) 606-617.

[18] Yang J and Zhang Y 2011 Alternating direction algorithms for $l_1$-problems in compressive sensing *SIAM Journal on Scientific Computing* 33 (1-2) 250-278.

[19] Georghiades A, Belhumeur P and Kriegman D 2001 From few to many: Illumination cone models for face recognition under variable lighting and pose *IEEE Transaction on Pattern Analysis and Machine Intelligence* 23 (6) 643-660.

[20] Tron R and Vidal R 2007 A benchmark for the comparison of 3-d motion segmentation algorithms *IEEE Conference on Computer Vision and Pattern Recognition* pp 1-8.

[21] Liu G, Lin Z, Yu Y 2010 Robust subspace segmentation by low-rank representation *International Conference on Machine Learning* pp 663-670.

[22] He X, Yan S, Hu Y and Niyogi P 2005 Face recognition using laplacianfaces *IEEE Transaction on Pattern Analysis and Machine Intelligence* 27 (3) 328-340.