Scaled Simplex Representation for Subspace Clustering

Jun Xu, Mengyang Yu, Ling Shao, *Senior Member, IEEE*, Wangmeng Zuo, *Senior Member, IEEE*, Deyu Meng, Lei Zhang, *Fellow, IEEE*, and David Zhang, *Fellow, IEEE*

***Abstract*—The self-expressive property of data points, that is, each data point can be linearly represented by the other data points in the same subspace, has proven effective in leading subspace clustering (SC) methods. Most self-expressive methods usually construct a feasible affinity matrix from a coefficient matrix, obtained by solving an optimization problem. However, the negative entries in the coefficient matrix are forced to be positive when constructing the affinity matrix via exponentiation, absolute symmetrization, or squaring operations. This consequently damages the inherent correlations among the data. Besides, the affine constraint used in these methods is not flexible enough for practical applications. To overcome these problems, in this article, we introduce a scaled simplex representation (SSR) for the SC problem. Specifically, the non-negative constraint is used to make the coefficient matrix physically meaningful, and the coefficient vector is constrained to be summed up to a scalar $s < 1$ to make it more discriminative. The proposed SSR-based SC (SSRSC) model is reformulated as a linear equality-constrained problem, which is solved efficiently under the alternating direction method of multipliers framework. Experiments on benchmark datasets demonstrate that the proposed SSRSC algorithm is very efficient and outperforms the state-of-the-art SC methods on accuracy. The code can be found at https://github.com/csjunxu/SSRSC.**

***Index Terms*—Scaled simplex representation (SSR), self-expressiveness, subspace clustering (SC).**

Manuscript received March 30, 2019; revised August 3, 2019; accepted September 23, 2019. This work was supported in part by the Major Project for New Generation of AI under Grant 2018AAA0100400, in part by the Natural Science Foundation of China under Grant 61572264, Grant 61620106008, Grant 61802324, and Grant 61772443, and in part by the Tianjin Natural Science Foundation under Grant 17JCJQJC43700 and Grant 18ZXZNGX00110. This article was recommended by Associate Editor R. Tagliaferri. *(Corresponding author: Jun Xu.)*

J. Xu is with the College of Computer Science, Nankai University, Tianjin 300071, China, and also with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE (e-mail: nankaimathxujun@gmail.com).

M. Yu and L. Shao are with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE.

W. Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China.

D. Meng is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China.

L. Zhang is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong.

D. Zhang is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, and also with the School of Science and Engineering, Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2943691

I. Introduction

IMAGE-PROCESSING problems often contain high-dimensional data, whose structure typically lie in a union of low-dimensional subspaces [1]. Recovering these low-dimensional subspaces from the high-dimensional data can reduce the computational and memory cost of the subsequent algorithms. To this end, many image-processing tasks require to clustering high-dimensional data in a way that each cluster can be fitted by a low-dimensional subspace. This problem is known as subspace clustering (SC) [1].

SC has been extensively studied over the past few decades [2]–[47]. Most existing SC methods fall into four categories: 1) iterative methods [2], [3]; 2) algebraic methods [4], [5], [14]; 3) statistical methods [6]–[9]; and 4) self-expressive methods [10]–[13], [15]–[47]. Among these methods, self-expressive ones are the most widely studied due to their theoretical soundness and promising performance in real-world applications, such as motion segmentation [16], face clustering [18], and digit clustering [44]. Self-expressive methods usually follow a three-step framework.

Step 1) A coefficient matrix is obtained for the data points by solving an optimization problem.

Step 2) An affinity matrix is constructed from the coefficient matrix by employing exponentiation [13]; absolute symmetrization [15], [16], [29], [31]–[39], [44]; squaring operations [17]–[24], [26]–[28], [40], [42]; etc.

Step 3) Spectral techniques [48] are applied to the affinity matrix and the final clusters are obtained for the data points.

Self-expressive methods [13], [15]–[24], [26]–[34], [36]–[47] obtain the coefficient matrix under the *self-expressiveness* property [15]: each data point in a union of multiple subspaces can be *linearly* represented by the other data points in the same subspace. In real-world applications, data points often lie in a union of multiple affine rather than linear subspaces [16] and, hence, the affine constraint is introduced [15], [16] to constrain the sum of coefficients to be 1. However, most self-expressive methods [13], [15]–[18], [28]–[34], [36]–[44], [46], [47] suffer from three major drawbacks. First, negative coefficients cannot be explicitly avoided in these methods in step 1. But it is physically problematic to reconstruct a real data point by allowing the others to "cancel each other out" with complex additions and subtractions [49]. Second, under the affine constraint, the coefficient vector is not flexible enough to handle the real-world cases where the

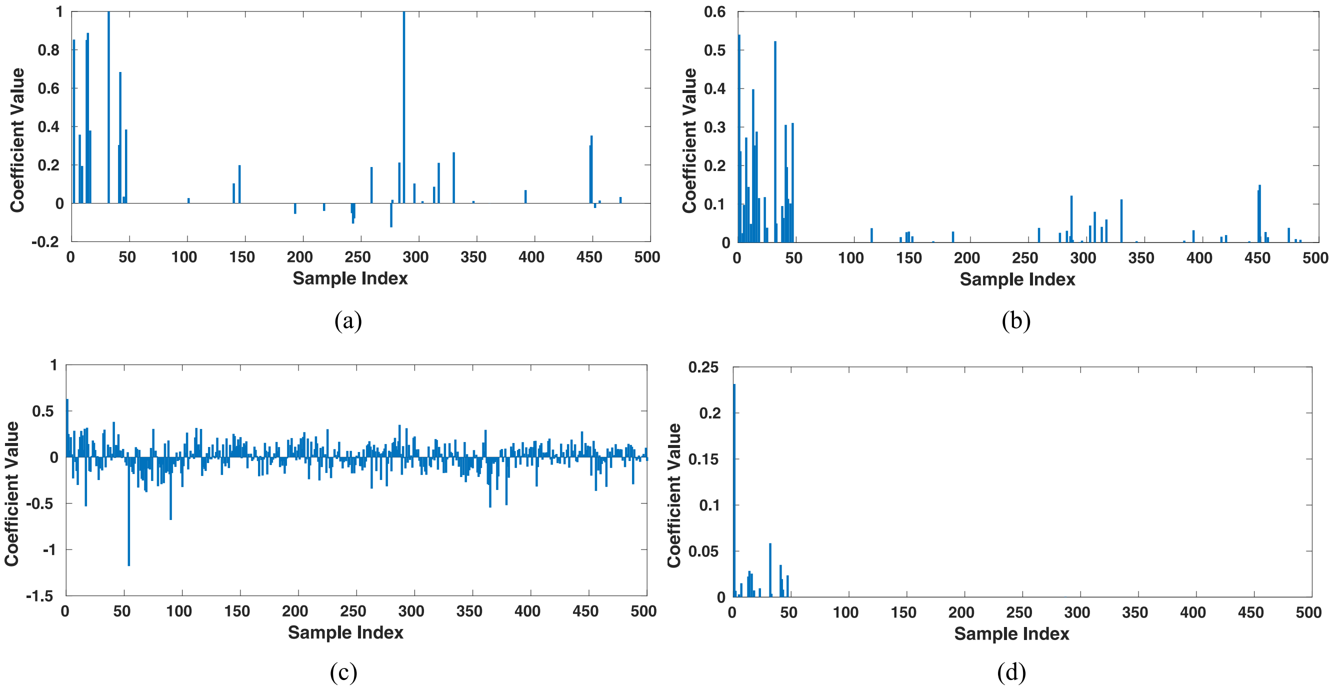

Fig. 1. Comparison of coefficient vectors by different representation schemes on the handwritten digit images from MNIST [50]. A digit sample "0" is used to compute the vector over 500 samples of digits $\{0, 1, \ldots, 9\}$ (50 for each). (a) Vectors solved by a sparse model (e.g., LASSO [51]) are confusing. (b) Vectors solved by the LSR model with non-negative constraints are noisy. (c) Coefficients solved by LSR with affine constraints are chaotic. (d) LSRs with the proposed SSR can obtain physically more meaningful coefficients vectors.

data points are often corrupted by noise or outliers. Third, the exponentiation, absolute symmetrization, or squaring operations in step 2 force the negative coefficients to be positive; thus, damaging the inherent correlations among the data points.

To solve the three drawbacks mentioned above, we propose a scaled simplex representation (SSR) for self-expressive-based SC. Specifically, we first extend the affine constraint to the scaled version in the optimization model, and the coefficient vector will sum up to a scalar s $(0 < s < 1)$ instead of 1. By tuning s, we can flexibly alter the generative and discriminative properties of the proposed SSR. Second, we utilize a non-negative constraint to make the representation more physically meaningful. By introducing that the non-negative constraint, it has three primary benefits.

1) In step 1, it eliminates the physically problematic subtractions among data points in optimization.
2) The obtained coefficient matrix in step 1 maintains the inherent correlations among data points by avoiding exponentiation, absolute symmetrization, or squaring operations when constructing the affinity matrix in step 2.
3) The non-negativity could potentially enhance the discriminability of the coefficients [52] with the scaled affine constraint such that a data point is more likely reconstructed by the data points from the same subspace.

To illustrate the advantages of the proposed SSR, in Fig. 1, we show the comparison of coefficient vectors solved by different representation schemes on the handwritten digit images from MNIST [50]. A digit sample "0" is used to compute the vector over 500 samples of digits $\{0, 1, \ldots, 9\}$ (50 for each). We observe that: the vector solved by a sparse model (e.g., LASSO [51]) is confusing, and the coefficients over the samples of other digits are also nonzero [Fig. 1(a)]. The vector solved by the least-square regression (LSR) model with non-negative constraint is noisy [Fig. 1(b)]. The coefficients solved by LSR with affine constraint are densely distributed over all samples [Fig. 1(c)]. The LSR with the proposed SSR can obtain a physically more meaningful coefficients vector [Fig. 1(d)].

With the introduced SSR, we propose a novel SSR-based SC (SSRSC) model. The experimental results on several benchmark datasets demonstrate that the proposed SSRSC achieves better performance than the state-of-the-art algorithms. In summary, our contributions are three-fold.

1) We introduce a new SSR for SC. The proposed SSR cannot only maintain the inherent correlations among the data points but also handle practical problems more flexibly.
2) We propose an SSRSC model for SC. We reformulate the proposed SSRSC model into a linear equality-constrained problem with two variables and solve the problem by an alternating direction method of multipliers (ADMMs) [53]. Each variable can be updated efficiently, and the convergence can be guaranteed.
3) We performed comprehensive experiments on several benchmark datasets, that is, Hopkins 155 [54], ORL, Extended Yale B [55], MNIST [50], and EMNIST. The results demonstrate that the proposed SSRSC algorithm is very efficient and achieves better performance than the state-of-the-art SC algorithms on motion segmentation, face clustering, and handwritten digits/letters clustering.

The remainder of this article is organized as follows. In Section II, we briefly survey the related work. In Section III, we first present the proposed SSRSC model, then provide its optimization, and finally, present the proposed SSRSC-based SC algorithm. Extensive experiments are conducted in Section IV to evaluate the SSRSC algorithm and compare it with the state-of-the-art SC algorithms. The conclusion and future work are given in Section V.

II. Related Work

A. Prior Work on Subspace Clustering

According to the employed mathematical framework, most existing SC algorithms [2]–[18], [28]–[34], [36]–[44], [46], [47], [56] can be divided into four main categories: 1) iterative methods; 2) algebraic methods; 3) statistical methods; and 4) self-expressive methods. To make this section as compact as possible, please refer to [16] for the introduction of iterative methods, algebraic methods, and statistical methods. Self-expressive methods are closely related to this article. They usually use local information around each data point to compare the similarity of two points. Inspired by the compressed sensing theory [57], [58], sparse SC (SSC) [15] solves the clustering problem by seeking a sparse representation of data points over themselves. By resolving the sparse representations for all data points and constructing an affinity graph, SSC automatically finds different subspaces as well as their dimensions from a union of subspaces. A robust version of SSC that deals with noise, corruptions, and missing observations is given in [16]. Instead of finding a sparse representation, the low-rank representation (LRR) method [17], [18], [28] poses the SC problem as finding an LRR of the data over themselves. Lu *et al.* proposed a clustering method based on LSR [31] to take advantage of data correlations and group highly correlated data together. The grouping information can be used to construct an affinity matrix, that is, block diagonal and can be used for SC through spectral clustering algorithms. Recently, Lin *et al.* analyzed the grouping effect in depth and proposed a smooth representation (SMR) framework [32] which also achieves state-of-the-art performance for the SC problem. Different from SSC, the LRR, LSR, and SMR algorithms all use normalized cuts [59] in the spectral clustering step. You *et al.* proposed a scalable orthogonal matching pursuit (OMP) method [44] to solve the SSC model [16]. You *et al.* also developed an elastic-net SC (EnSC) model [45] which correctly predicted relevant connections among different clusters. Ji *et al.* [41] developed the first unsupervised network for SC by learning the *self-expressiveness* property [15].

B. Self-Expressiveness-Based Framework

Most state-of-the-art SC methods are designed under the self-expressive framework. Mathematically, denoting the data matrix as $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$, each data point $\boldsymbol{x}_i \in \mathbb{R}^D$, $i \in \{1, \ldots, N\}$ in $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{x}_i = \boldsymbol{X}\boldsymbol{c}_i \tag{1}$$

where $\boldsymbol{c}_i \in \mathbb{R}^N$ is the coefficient vector. If the data points are listed column by column, (1) can be rewritten as

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{C} \tag{2}$$

where $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ is the coefficient matrix. To find the desired $\boldsymbol{C}$, the existing SC methods [15]–[18], [29], [31]–[34], [36]–[40], [44] impose various regularizations, such as sparsity and low rankness. Below, $\|\cdot\|_F$, $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_{2,1}$, $\|\cdot\|_*$, λ, and p denote the Frobenius norm, the ℓ_1-norm, the ℓ_2-norm, the $\ell_{2,1}$-norm, the nuclear norm, the regularization parameter, and a positive integer, respectively. The optimization models of several representative works are summarized as follows:

SSC [16] employs sparse and affine constraints

$$\min_{\boldsymbol{C}} \|\boldsymbol{C}\|_1 \quad \text{s.t.} \quad \boldsymbol{X} = \boldsymbol{X}\boldsymbol{C}, \boldsymbol{1}^\top\boldsymbol{C} = \boldsymbol{1}^\top, \operatorname{diag}(\boldsymbol{C}) = \boldsymbol{0}. \tag{3}$$

LRR [18] employs a low-rank constraint

$$\min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_{2,1} + \lambda\|\boldsymbol{C}\|_*. \tag{4}$$

LSR [31] is a vanilla least-square regression

$$\min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2 \quad \text{s.t.} \quad \operatorname{diag}(\boldsymbol{C}) = \boldsymbol{0}. \tag{5}$$

SSCOMP [44] is the SSC model solved by OMP

$$\min_{\boldsymbol{c}_i} \|\boldsymbol{x}_i - \boldsymbol{X}\boldsymbol{c}_i\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{c}_i\|_0 \le p, \boldsymbol{c}_{ii} = 0. \tag{6}$$

Once the coefficient matrix $\boldsymbol{C}$ is computed, the affinity matrix $\boldsymbol{A}$ is usually constructed by exponentiation [13]; absolute symmetrization [15], [16], [29], [31]–[34], [36]–[39], [44]; squaring operations [17], [18], [40], etc. For example, the widely used absolute symmetrization operation in [15], [16], [29], [31]–[34], [36]–[39], and [44] is defined by

$$\boldsymbol{A} = \Big(|\boldsymbol{C}| + |\boldsymbol{C}^\top|\Big)/2. \tag{7}$$

After the affinity matrix $\boldsymbol{A}$ is obtained, spectral clustering techniques [59] are applied to obtain the final segmentation of the subspaces. However, these self-expressive methods suffer from one major drawback: the exponentiation, absolute symmetrization, and squaring operations will force the negative entries in $\boldsymbol{C}$ to be positive in $\boldsymbol{A}$ and, hence, damage the inherent correlations among the data points in $\boldsymbol{X}$. Besides, the affine constraint in SSC limits its flexibility, making it difficult to deal with complex real-world applications. In order to remedy these drawbacks, in this article, we introduce the SSR to tackle the SC problem.

C. Other Constraints for Subspace Clustering

Though sparse [15], [16] or low-rank [17], [18], [60] representation and ridge regression [31], [61] are widely used in SC and other vision tasks [62]–[66], there are also other constraints employed by the existing SC algorithms. Feng *et al.* [34] proposed the Laplacian constraint for a block-diagonal matrix pursuit. The developed block-diagonal SSC and LRR show clear improvements over SSC and LRR on SC and graph construction for semisupervised learning. The log-determinant function is utilized in [67] as an alternative to the nuclear norm for low-rank constraint. The non-negative constraint has also been adopted in some recent works for similarity graph learning [68]–[72]. In these works, the non-negativity is employed to construct a sparse or low-rank similarity graph. This article shares the same spirit with these works on this point and aims to build a physically reasonable affinity matrix by employing non-negativity. Recently, Li *et al.* [73] proposed to directly learn the affinity matrix by the diffusion process to spread the local manifold structure of data along with their global manifolds. This article can be viewed as propagating the data manifold constraint into sparsity models to enhance its connectivity for SC. The simplex sparse constraint in [74] is closely related to this article. But our proposed scaled simplex constraint allows the sum of coefficients to be a scalar, while, in this article, the sum is fixed to be 1. Besides, to make our model simpler, we do not use the diagonal constraint in [74].

III. Simplex Representation-Based Subspace Clustering

In this section, we propose an SSRSC model, develop an optimization algorithm to solve it, and present a novel SSRSC-based algorithm for SC.

A. Proposed SSRSC Model

Given a data matrix $\boldsymbol{X}$, for each data point $\boldsymbol{x}_i$ in $\boldsymbol{X}$, our SSRSC model aims to obtain its coefficient vector $\boldsymbol{c}_i$ over $\boldsymbol{X}$ under the scaled simplex constraint $\{\boldsymbol{c}_i \geq 0, \mathbf{1}^\top \boldsymbol{c}_i = s\}$. Here, we employ the LSR as the objective function due to its simplicity. The proposed SSRSC model is formulated as follows:

$$\min_{\boldsymbol{c}_i} \|\boldsymbol{x}_i - \boldsymbol{X}\boldsymbol{c}_i\|_2^2 + \lambda\|\boldsymbol{c}_i\|_2^2 \text{s.t.} \quad \boldsymbol{c}_i \geq 0, \mathbf{1}^\top \boldsymbol{c}_i = s \tag{8}$$

where $\mathbf{1}$ is the vector of all ones and $s > 0$ is a scalar denoting the sum of entries in the coefficient vector $\boldsymbol{c}_i$. We use the term "scaled simplex" here because the entries in the coefficient vector $\boldsymbol{c}_i$ are constrained by a scaled simplex, that is, they are non-negative and sum up to a scalar s.

We can also rewrite the SSR-based model (8) for all N data points in the matrix form

$$\begin{aligned} \min_{\boldsymbol{C}} \quad & \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{C} \geq 0, \mathbf{1}^\top \boldsymbol{C} = s\mathbf{1}^\top \end{aligned} \tag{9}$$

where $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ is the coefficient matrix. Here, the constraint $\boldsymbol{C} \geq 0$ favors positive values for entries corresponding to data points from the same subspace, while suppressing entries corresponding to data points from different subspaces, thus making the coefficient matrix $\boldsymbol{C}$ discriminative. The constraint $\mathbf{1}^\top \boldsymbol{C} = s\mathbf{1}^\top$ limits the sum of each coefficient vector $\boldsymbol{c}_i$ to be s, thus making the representation more discriminative since each entry should be non-negative.

B. Model Optimization

The proposed SSRSC model (9) cannot be solved analytically. In this section, we solve it by employing variable splitting methods [75], [76]. Specifically, we introduce an auxiliary variable $\boldsymbol{Z}$ into the SSRSC model (9) and reformulate it as a linear equality-constrained problem

$$\begin{aligned} \min_{\boldsymbol{C},\boldsymbol{Z}} \quad & \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{Z}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{Z} \geq 0, \mathbf{1}^\top \boldsymbol{Z} = s\mathbf{1}^\top, \boldsymbol{Z} = \boldsymbol{C} \end{aligned} \tag{10}$$

whose solution with respect to $\boldsymbol{C}$ coincides with the solution of (9). Since the objective function in (10) is separable with respect to the variables $\boldsymbol{C}$ and $\boldsymbol{Z}$, it can be solved using the ADMM [53]. The corresponding augmented Lagrangian function is

$$\begin{aligned} &\mathcal{L}(\boldsymbol{C}, \boldsymbol{Z}, \boldsymbol{\Delta}, \rho) \\ &\quad = \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{Z}\|_F^2 + \langle \boldsymbol{\Delta}, \boldsymbol{Z} - \boldsymbol{C}\rangle + \frac{\rho}{2}\|\boldsymbol{Z} - \boldsymbol{C}\|_F^2 \\ &\quad = \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{Z}\|_F^2 + \frac{\rho}{2}\left\|\boldsymbol{Z} - \boldsymbol{C} + \frac{1}{\rho}\boldsymbol{\Delta}\right\|_F^2 \\ &\quad = \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \frac{2\lambda+\rho}{2}\left\|\boldsymbol{Z} - \frac{\rho}{2\lambda+\rho}\left(\boldsymbol{C} - \frac{1}{\rho}\boldsymbol{\Delta}\right)\right\|_F^2 \\ &\qquad + \frac{\lambda\rho}{2\lambda+\rho}\left\|\boldsymbol{C} - \frac{1}{\rho}\boldsymbol{\Delta}\right\|_F^2 \end{aligned} \tag{11}$$

where $\boldsymbol{\Delta}$ is the augmented Lagrangian multiplier and $\rho > 0$ is the penalty parameter. Denote by $(\boldsymbol{C}_k, \boldsymbol{Z}_k)$ and $\boldsymbol{\Delta}_k$, the optimization variables and Lagrange multiplier at iteration k ($k = 0, 1, 2, \ldots,$), respectively. We initialize the variables $\boldsymbol{C}_0$, $\boldsymbol{Z}_0$, and $\boldsymbol{\Delta}_0$ to be conformable zero matrices. By taking the derivatives of the Lagrangian function $\mathcal{L}$ with respect to $\boldsymbol{C}$ and $\boldsymbol{Z}$, and setting them to be zeros, we can alternatively update the variables as follows.

1) Updating $\boldsymbol{C}$ While Fixing $\boldsymbol{Z}_k$ and $\boldsymbol{\Delta}_k$

$$\boldsymbol{C}_{k+1} = \arg\min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \frac{\rho}{2}\left\|\boldsymbol{C} - \left(\boldsymbol{Z}_k + \frac{1}{\rho}\boldsymbol{\Delta}_k\right)\right\|_F^2. \tag{12}$$

This is a standard LSR problem with a closed-form solution

$$\boldsymbol{C}_{k+1} = \left(\boldsymbol{X}^\top\boldsymbol{X} + \frac{\rho}{2}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{X}^\top\boldsymbol{X} + \frac{\rho}{2}\boldsymbol{Z}_k + \frac{1}{2}\boldsymbol{\Delta}_k\right). \tag{13}$$

We note that the complexity for updating $\boldsymbol{C}$ is $\mathcal{O}(N^3)$ when there are N data points in the data matrix $\boldsymbol{X}$. This cube complexity largely hinders the practical usage of the proposed method. In order to improve the speed (while maintaining the accuracy) of SSRSC, we employ the Woodbury formula [77] to compute the inversion in (13) as

$$\left(\boldsymbol{X}^\top\boldsymbol{X} + \frac{\rho}{2}\boldsymbol{I}\right)^{-1} = \frac{2}{\rho}\boldsymbol{I} - \left(\frac{2}{\rho}\right)^2\boldsymbol{X}^\top\left(\boldsymbol{I} + \frac{2}{\rho}\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1}\boldsymbol{X}. \tag{14}$$

By this step, the complexity of updating $\boldsymbol{C}$ is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(DN^2)$. Since $(\boldsymbol{X}^\top\boldsymbol{X} + (\rho/2)\boldsymbol{I})^{-1}$ is not updated during iterations, we can also precompute it and store it before iterations. This further saves the abundant computational costs.

2) Updating $\boldsymbol{Z}$ While Fixing $\boldsymbol{C}_k$ and $\boldsymbol{\Delta}_k$

$$\begin{aligned} \boldsymbol{Z}_{k+1} = \arg\min_{\boldsymbol{Z}} \quad & \left\|\boldsymbol{Z} - \frac{\rho}{2\lambda+\rho}\left(\boldsymbol{C}_{k+1} - \rho^{-1}\boldsymbol{\Delta}_k\right)\right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{Z} \geq 0, \mathbf{1}^\top \boldsymbol{Z} = s\mathbf{1}^\top. \end{aligned} \tag{15}$$

This is a quadratic programming problem with a strictly convex objective function and a close and convex constraint, so there is a unique solution. As such, problem (15) can be solved using, for example, active set methods [78], [79] or projection-based methods [80]–[82]. Here, we employ the projection-based method [81], whose computational complexity is $\mathcal{O}(N\log N)$ to project a vector of length N onto a

Algorithm 1 Projection of Vector $\boldsymbol{u}_{k+1}$ Onto a Simplex

```
Input: Data point $\boldsymbol{u}_{k+1} \in \mathbb{R}^N$, scalar $s$;
1. Sort $\boldsymbol{u}_{k+1}$ into $\boldsymbol{w}$: $w_1 \geq w_2 \geq \cdots \geq w_N$;
2. Find $\alpha = \max\{1 \leq j \leq N: w_j + \frac{1}{j}(s - \sum_{i=1}^{j} w_i) > 0\}$;
3. Define $\beta = \frac{1}{\alpha}(s - \sum_{i=1}^{\alpha} w_i)$;
Output: $z_{k+1}$: $z_{k+1}^i = \max\{u_{k+1}^i + \beta, 0\}$, $i = 1, \ldots, N$.
```

Algorithm 2 Solve the SSRSC Model (10) via ADMM

```
Input: Data matrix $\boldsymbol{X}$, Tol $> 0$, $\rho > 0$, $K$;
Initialization: $\boldsymbol{C}_0 = \boldsymbol{Z}_0 = \boldsymbol{\Delta}_0 = \boldsymbol{0}$, T = False, $k = 0$;
While (T == False) do
1. Update $\boldsymbol{C}_{k+1}$ by Eqn. (13);
2. Update $\boldsymbol{Z}_{k+1}$ by Eqn. (15);
3. Update $\boldsymbol{\Delta}_{k+1}$ by Eqn. (16);
4. if (Convergence condition is satisfied) or ($k \geq K$)
5.    T ← True;
   end if
end while
Output: Matrices $\boldsymbol{C}$ and $\boldsymbol{Z}$.
```

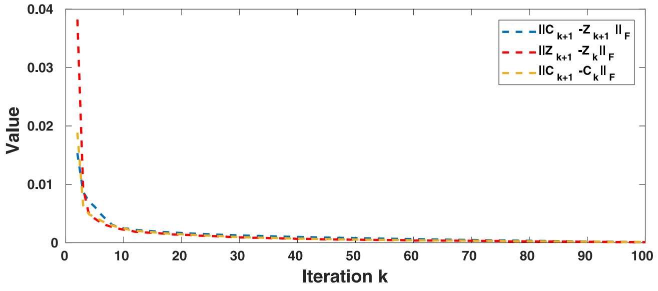

Fig. 2. Convergence curves of $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F$ (red line), $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F$ (blue line), and $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F$ (green line) of the proposed SSRSC on the "1R2RC" sequence from the Hopkins155 dataset [54].

simplex. Denoting by $\boldsymbol{u}_{k+1}$, an arbitrary column of $[\rho/(2\lambda + \rho)](\boldsymbol{C}_{k+1} - \rho^{-1}\boldsymbol{\Delta}_k)$, the solution of z_{k+1} (the corresponding column in $\boldsymbol{Z}_{k+1}$) can be solved by projecting $\boldsymbol{u}_{k+1}$ onto a simplex [81]. The solution of problem (15) is summarized in Algorithm 1.

3) Updating $\boldsymbol{\Delta}$ *While Fixing* $\boldsymbol{C}_k$ *and* $\boldsymbol{Z}_k$

$$\boldsymbol{\Delta}_{k+1} = \boldsymbol{\Delta}_k + \rho(\boldsymbol{Z}_{k+1} - \boldsymbol{C}_{k+1}). \tag{16}$$

We repeat the above alternative updates until a certain convergence condition is satisfied or the number of iterations reaches a preset threshold K. Under the convergence condition of the ADMM algorithm, $\|\boldsymbol{C}_{k+1} - \boldsymbol{Z}_{k+1}\|_F \leq$ Tol, $\|\boldsymbol{C}_{k+1} - \boldsymbol{C}_k\|_F \leq$ Tol, and $\|\boldsymbol{Z}_{k+1} - \boldsymbol{Z}_k\|_F \leq$ Tol must be simultaneously satisfied, where Tol $= 0.01$ tolerates small errors. Since the objective function and constraints are both convex, problem (10) solved by ADMM is guaranteed to converge at a global optimal solution. We summarize these updating procedures in Algorithm 2.

Convergence nalysis: The convergence of Algorithm 2 can be guaranteed since the overall objective function (10) is convex with a global optimal solution. In Fig. 2, we plot the convergence curves of the errors of $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F$, $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F$, and $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F$. One can see that they are reduced to less than Tol $= 0.01$ simultaneously in five iterations.

C. Theoretical Analysis

Our optimization problem (9) is a convex optimization problem, which means $f(\boldsymbol{C}) := \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2$ is a convex function and the set $\mathcal{S} := \{\boldsymbol{C}|\boldsymbol{C} \geq 0, \mathbf{1}^\top\boldsymbol{C} = s\mathbf{1}^\top\}$ is a convex set. Then, the function f has a minimum on the hyperplane $\overline{\mathcal{S}} := \{\boldsymbol{C}|\mathbf{1}^\top\boldsymbol{C} = s\mathbf{1}^\top\}$, which is the linear span of $\mathcal{S}$. Below, we discuss the solution space of f over $\mathcal{S}$.

Theorem 1: Suppose $\boldsymbol{C}^*$ is the minimum of the convex function f over the convex set $\mathcal{S}$. If $\boldsymbol{C}^*$ is not the minimum of f on $\overline{\mathcal{S}}$, then $\boldsymbol{C}^*$ is on the boundary of $\mathcal{S}$: $\partial\mathcal{S}$.

Proof: We first make an assumption that $\boldsymbol{C}^* \notin \partial\mathcal{S}$, then we will derive a contradiction. If our assumption holds, there exists a high-dimensional open sphere $B_r(\boldsymbol{C}^*) = \{\boldsymbol{C}|\|\boldsymbol{C} - \boldsymbol{C}^*\| < r\} \subset \mathbb{R}^{N\times N}$ centered at $\boldsymbol{C}^* \cap \mathcal{S}$ with radius $r > 0$, such that for all $\boldsymbol{C} \in B_r$, $f(\boldsymbol{C}) \geq f(\boldsymbol{C}^*)$ holds.

We consider $\forall \boldsymbol{D} \in \overline{\mathcal{S}}$, $\exists\lambda \in (0, 1)$ such that $\boldsymbol{C}^* + \lambda(\boldsymbol{D} - \boldsymbol{C}^*) \in B_r(\boldsymbol{C}^*)$. In fact, we can set $\lambda < \min([r/(\|\boldsymbol{D} - \boldsymbol{C}^*\|)], 1)$. Then, we have $\|\boldsymbol{C}^* + \lambda(\boldsymbol{D} - \boldsymbol{C}^*) - \boldsymbol{C}^*\| = \|\lambda(\boldsymbol{D} - \boldsymbol{C}^*)\| < r$, which means $\boldsymbol{C}^* + \lambda(\boldsymbol{D} - \boldsymbol{C}^*) \in B_r(\boldsymbol{C}^*)$. In this way, we have

$$f\big(\boldsymbol{C}^* + \lambda\big(\boldsymbol{D} - \boldsymbol{C}^*\big)\big) \geq f\big(\boldsymbol{C}^*\big). \tag{17}$$

However, due to the convexity of f, we have the following Jensen's inequality:

$$\begin{aligned} f\big(\boldsymbol{C}^* + \lambda\big(\boldsymbol{D} - \boldsymbol{C}^*\big)\big) &= f\big((1 - \lambda)\boldsymbol{C}^* + \lambda\boldsymbol{D}\big) \\ &\leq (1 - \lambda)f\big(\boldsymbol{C}^*\big) + \lambda f(\boldsymbol{D}). \end{aligned} \tag{18}$$

Combining this with (17), we obtain $\lambda f(\boldsymbol{D}) \geq \lambda f(\boldsymbol{C}^*)$ and, hence, $f(\boldsymbol{D}) \geq f(\boldsymbol{C}^*)$, $\forall\boldsymbol{D} \in \mathbb{R}^{N\times N}$. Then, $\boldsymbol{C}^*$ is the minimum of f on the hyperplane $\overline{\mathcal{S}}$, resulting in a contradiction. ■

D. Discussion

The constraint "$\mathbf{1}^\top\boldsymbol{c} = 1$" has already been used in SSC [15], [16] to deal with the presence of affine, rather than linear subspaces. However, limiting the sum of the coefficient vector $\boldsymbol{c}$ to be 1 is not flexible enough for real-world clustering problems. What is more, suppressing the sum of the entries in the coefficient vector $\boldsymbol{c}$ can make it more discriminative, since these entries should be non-negative and sum up to a scalar s. Considering the extreme case where s is nearly zero, each data point must be represented by its most similar data points in the homogeneous subspace. To this end, in our proposed SSRSC model, we extend the affine constraint of "summing up to 1" to a scaled affine constraint of "summing up to a scalar s." In our experiments (please refer to Section IV), we observe the improved performance of SSRSC on SC by this extension.

In real-world applications, data are often corrupted by outliers due to *ad-hoc* data-collection techniques. Existing SC methods deal with outliers by explicitly modeling them as an additional variable, and updating this variable using the ADMM algorithm. For example, in the seminal work of SSC [15], [16] and its successors [31], [33], [37], [39], c_{ii} is set as 0 for $\boldsymbol{x}_i$, indicating that each data point cannot be represented by itself, thus avoiding the trivial

```
Algorithm 3 SC by SSRSC
Input: A set of data points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ lying in
    a union of $n$ subspaces $\{\mathcal{S}_j\}_{j=1}^n$;
1. Obtain the coefficient matrix $\boldsymbol{C}$ by solving the SSRSC model:
    $\min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2$ s.t. $\boldsymbol{C} \geq 0, \boldsymbol{1}^\top\boldsymbol{C} = s\boldsymbol{1}^\top$;
2. Construct the affinity matrix by
    $\boldsymbol{A} = \frac{\boldsymbol{C}+\boldsymbol{C}^\top}{2}$;
3. Apply spectral clustering [83] to the affinity matrix;
Output: Segmentation of data: $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$.
```

solution of identity matrix. However, this brings additional computational costs and prevents the entire algorithm from converging [16], [18]. Different from these existing methods [16], [31], [33], [37], [39], we do not consider the constraint of $c_{ii} = 0$ for three major reasons. First, the positive λ in the regularization term can naturally prevent the trivial solution of identity matrix $\boldsymbol{C} = \boldsymbol{I}$. Second, $c_{ii} \neq 0$ has a clear physical meaning, allowing a sample $\boldsymbol{x}_i$ in the subspace to be *partially* represented by itself. This is particularly useful when $\boldsymbol{x}_i$ is corrupted by noise. By removing the constraint of $c_{ii} = 0$, our proposed SSRSC model is more robust to noise, as will be demonstrated in the ablation study in Section IV-D.

E. Subspace Clustering via Simplex Representation

SC Algorithm: Denote by $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$, a set of data points drawn from a union of n subspaces $\{\mathcal{S}_j\}_{j=1}^n$. Most existing spectral clustering-based SC algorithms [16], [18], [29], [32], [37], [44] first compute the coefficient matrix $\boldsymbol{C}$, and then construct a non-negative affinity matrix $\boldsymbol{A}$ from $\boldsymbol{C}$ by exponentiation [13]; absolute symmetrization [15], [16], [29], [31]–[34], [36], [37], [39], [44]; squaring operations [17], [18], [40]; etc. In contrast, in our proposed SSRSC model, the coefficient matrix is guaranteed to be non-negative by the introduction of a simplex constraint. Hence, we can remove the absolute operation and construct the affinity matrix by

$$\boldsymbol{A} = \left(\boldsymbol{C} + \boldsymbol{C}^\top\right)/2. \tag{19}$$

As a common postprocessing step in [15], [16], [29], [17], [18], [31]–[34], [36], [37], [39], [40], and [44], we apply the spectral clustering technique [83] to the affinity matrix $\boldsymbol{A}$ and obtain the final segmentation of data points. Specifically, we employ the widely used normalized cut algorithm [59] to segment the affinity matrix. The proposed SSRSC-based SC algorithm is summarized in Algorithm 3.

Complexity Analysis: Assume that there are N data points in the data matrix $\boldsymbol{X}$. In Algorithm 2, the costs for updating $\boldsymbol{C}$ and $\boldsymbol{Z}$ are $\mathcal{O}(N^2D)$ and $\mathcal{O}(N^2 \log N)$, respectively. The costs for updating $\boldsymbol{\Delta}$ and ρ are negligible compared to the updating costs of $\boldsymbol{C}$ and $\boldsymbol{Z}$. As such, the overall complexity of Algorithm 2 is $\mathcal{O}(\max(D, \log N)N^2K)$, where K is the number of iterations. The costs for affinity matrix construction and spectral clustering in Algorithm 3 can be ignored. Hence, the overall cost of the proposed SSRSC is $\mathcal{O}(\max(D, \log N)N^2K)$ for data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times N}$.

IV. Experiments

In this section, we first compare the proposed SSRSC with the state-of-the-art SC methods. The comparison is performed on five benchmark datasets on motion segmentation for video analysis, human faces clustering, and handwritten digits/letters clustering. Then, we validate the effectiveness of the proposed scaled simplex constraints for SC through comprehensive ablation studies.

A. Implementation Details

The proposed SSRSC model (9) is solved under the ADMM [53] framework. There are four parameters to be determined in the ADMM algorithm: 1) the regularization parameter λ; 2) the sum s of the entries in the coefficient vector; and 3) the penalty parameter ρ, and the iteration number K. In all experiments, we fix $s = 0.5$, $\rho = 0.5$, and $K = 5$. As in most competing methods [13]–[18], [29], [31]–[34], [36]–[40], [44], parameter λ is tuned on each dataset to achieve the best performance of SSRSC on that dataset. The influence of λ on each dataset will be introduced in Section IV-C. All experiments are run under the MATLAB2014b environment on a machine with a CPU of 3.50 GHz and 12-GB RAM.

B. Datasets

We evaluate the proposed SSRSC method on the Hopkins155 dataset [54] for motion segmentation, the Extended Yale B [55] and ORL [84] datasets for human face clustering, and the MNIST [50] and EMNIST [85] for handwritten digits/letters clustering.

The *Hopkins155 dataset* [54] contains 155 video sequences, 120 of which contain two moving objects and 35 of which contain three moving objects, corresponding to 2 or 3 low-dimensional subspaces of the ambient space. On average, each two-motion sequence has 30 frames and each frame contains $N = 266$ data points, while each three-motion sequence has 29 frames and each frame contains $N = 393$ data points. Similar to the experimental settings in the previous methods, such as SSC [16], on this dataset, we employ principal component analysis (PCA) [86] to project the original trajectories of different objects into a 12-D subspace, in which we evaluate the comparison methods. This dataset [54] has been widely used as a benchmark to evaluate the SC methods for motion segmentation. Fig. 3 presents some segmentation examples from the Hopkins155 dataset [54], where different colors indicate different moving objects.

The *Extended Yale B dataset* [55] contains face images of 38 human subjects, and each subject has 64 near-frontal images (grayscale) taken under different illumination conditions. The original images are of size 192 × 168 pixels and we resize them to 48 × 42 pixels in our experiments. For dimension reduction purposes, the resized images are further projected onto a $6n$-dimensional subspace using PCA, where n is the number of subjects (or subspaces) selected in our experiments. Following the experimental settings in SSC [16], we divide the 38 subjects into 4 groups, consisting of subjects 1 to 10, subjects 11 to 20, subjects 21 to 30, and subjects 31 to 38. For each of the first three groups, we select $n \in \{2, 3, 5, 8, 10\}$

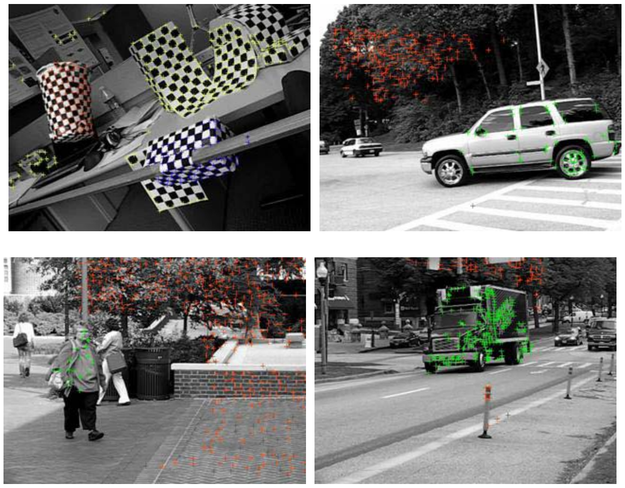

Fig. 3. Exemplar motion segmentation of one frame from different sequences in the Hopkins155 dataset [54].

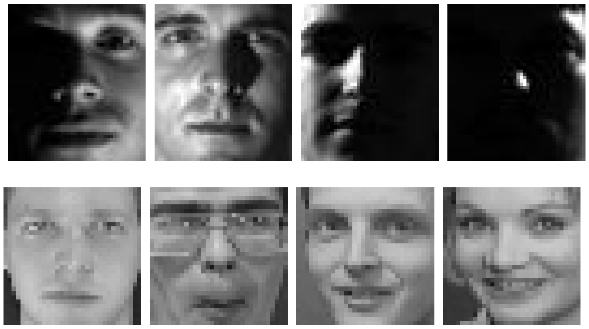

Fig. 4. Face images from the Extended Yale B dataset [55] (top) and the ORL dataset [84] (bottom).

subjects, while for the last group, we choose $n \in \{2, 3, 5, 8\}$. Finally, we apply SC algorithms for each set of n subjects. Fig. 4 (top) shows some face images from Extended Yale B, captured under different lighting conditions.

The *ORL dataset* [84] contains overall 400 human face images of 40 subjects, each having 10 samples. Similar to [41], we resize the original face images from 112×92 to 32×32. For each human subject, the face images were taken under varying lighting conditions, with different facial expressions (e.g., open eyes or closed eyes, smiling or not smiling, etc.), as well as different facial details (e.g., w/ glasses or w/o glasses). The **ORL** dataset is more difficult to tackle than the Extended Yale B [55] for two reasons: 1) its varying face images varies much more complex and 2) it only contains 400 images, much smaller than Extended Yale B (2432 images). Fig. 4 (bottom) shows some face images from ORL.

MNIST dataset [50] contains 60 000 grayscale images of 10 digits (i.e., $\{0, \ldots, 9\}$) in the training set and 10 000 images in the testing set. The images are of size 28×28 pixels. In our experiments, we randomly select $N_i \in \{50, 100, 200, 400, 600\}$ images for each of the 10 digits. Following [44], for each image, a set of feature vectors is computed using a scattering convolution network (SCN) [87]. The final feature vector is a concatenation of the coefficients in each layer of the network and is translation invariant and deformation stable. Each feature vector is 3472-D. The feature vectors for all images are then projected onto a 500-D subspace using PCA. Fig. 5 shows some examples of the handwritten digit images in this dataset.

Fig. 5. Digit images from the MNIST dataset [50].

EMNIST dataset [85] is an extension of the MNIST dataset that contains grayscale handwritten digits and letters. This dataset contains overall 190 998 images corresponding to 26 lowercase letters. We use them as the data for a 26-class clustering problem. The images are of size 28×28. In our experiments, we randomly select $N_i = 500$ images for each of the 26 digits/letters. Following [47], for each image, a set of feature vectors is computed using an SCN [87], which is translation invariant and deformation stable. The feature vectors are 3472-D, and the feature vectors for all images are projected onto a 500-D subspace using PCA.

C. Comparison With State-of-the-Art Methods

Comparison Methods: We compare the proposed SSRSC with several state-of-the-art SC methods, including SSC [15], [16]; LRR [17], [18]; LRSC [29]; LSR [31]; SMR [32]; S3C [37], [38]; RSIM [14]; SSCOMP [44]; EnSC [45]; DSC [41]; and ESC [47]. For SSRSC, we fix $s = 0.5$, or as reported in the ablation study, that is, $s = 0.9$ on Hopkins155, $s = 0.25$ on Extended YaleB, $s = 0.4$ on ORL, and $s = 0.15$ on MNIST. For the SMR method, we use the J_1 affinity matrix [i.e., (7)] in Section II as described in [32], for fair comparison. For the other methods, we tune their corresponding parameters on each of the three datasets, that is, the Hopkins155 dataset [54] for motion segmentation, the Extended Yale B dataset [55] for face clustering, and the MNIST dataset [50] for handwritten digit clustering, to achieve their best clustering results.

Comparison on Affinity Matrix: The affinity matrix plays a key role in the success of SC methods. Here, we visualize the affinity matrix of the proposed SSRSC and the comparison methods on the SC problem. We run the proposed SSRSC algorithm and the competing methods [14], [16], [18], [29], [31], [37], [38], [44], [45] on the MNIST dataset [50]. The training set contains 6000 images for each digit in $\{0, 1, \ldots, 9\}$. We randomly select 50 images for each digit, and use 500 total images to construct the affinity matrices using these competing SC methods. The results are visualized in Fig. 6. As can be seen, the affinity matrix of SSRSC shows better connections within each subspace and generates less noise than most of the other methods. Though LSR [31] and RSIM [14] have less noise than our proposed SSRSC, their affinity matrices suffer from strong diagonal entries, indicating that for these two methods, the data points are mainly reconstructed by themselves. With the scaled simplex constraint, the proposed SSRSC algorithm can better exploit the inherent correlations among the data points and achieve better clustering performance than the other compared methods.

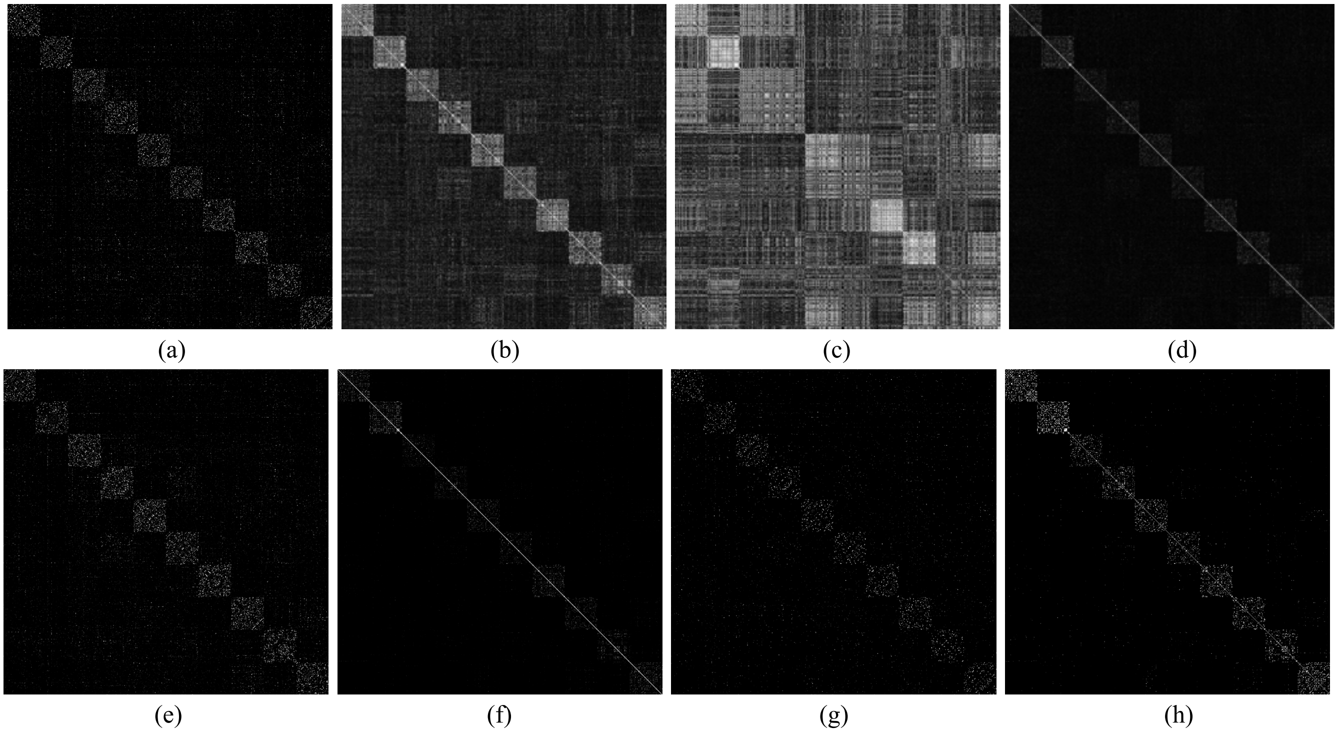

Fig. 6. Affinity matrices and corresponding average clustering errors by different methods on the handwritten digit images from MNIST [50]. Fifty images for each digit of $\{0, 1, \ldots, 9\}$ are used to compute the affinity matrix by different methods. All images are normalized to [0, 1] by dividing the maximal entries in the corresponding affinity matrices. (a) SSC [16]: 16.99%. (b) LRR [18]: 17.97%. (c) LRSC [29]: 24.16%. (d) LSR [31]: 24.98%. (e) S3C [38]: 15.92%. (f) RSIM [14]: 18.13%. (g) SSCOMP [44]: 16.36%. (h) SSRSC: 11.81%

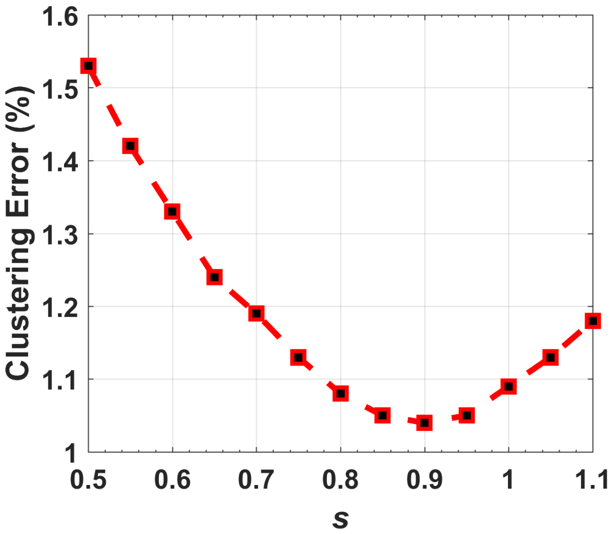

Fig. 7. Average clustering errors (%) of the proposed SSRSC algorithm with different scalar s on the Hopkins155 dataset [54].

Results on Motion Segmentation: We first study how parameter s influences the average clustering errors of the proposed SSRSC algorithm. The clustering errors with respect to the value of s are plotted in Fig. 7. As can be seen, the proposed SSRSC obtains an average clustering error of 1.53% when $s = 0.5$. Note that SSRSC achieves its lowest average clustering error of 1.04% when $s = 0.9$. The parameter λ is set as $\lambda = 0.001$. We then compare the proposed SSRSC with the other competing SC algorithms [14], [16], [18], [29], [31], [32], [37], [38], [44], [45]. The results on the average clustering errors are listed in Table I, from which we can see that the proposed SSRSC achieves the lowest clustering error. Besides, the speed of the proposed SSRSC approach is only slightly slower than LRSC, LSR, SSCOMP, and EnSC, and much faster than the other competing methods. Note that the fast speed of SSRSC is due to both the efficient solution in each iteration and fewer iterations in the ADMM algorithm.

Results on Human Face Clustering: Here, we compare the proposed SSRSC algorithm with the competing methods on the commonly used Extended Yale B dataset [55] and ORL dataset [84] for human face clustering. On ORL [84], we also compared the deep SC (DSC) method, which achieves state-of-the-art performance on this dataset.

We study how the scalar s influences the clustering errors (%) of the proposed SSRSC algorithm and takes the Extended Yale B dataset [55] for an example. The average clustering errors with respect to the value of s are plotted in Fig. 8. As can be seen, SSRSC achieves an average clustering error of 3.26% when $s = 0.5$ and achieves the lowest average clustering error of 2.16% when $s = 0.25$. We set $\lambda = 0.005$.

The comparison results of different algorithms are listed in Tables II and III. From Table II, we observe that for different numbers ($\{2, 3, 5, 8, 10\}$) of clustering subjects, the average clustering errors of SSRSC are always significantly lower than the other competing methods. For example, when clustering 10 subjects, the average error of SSRSC (when fixing $s = 0.5$) is 5.10%, while the errors of the other methods are 10.94% for

TABLE I
AVERAGE CLUSTERING ERRORS (%) AND SPEED (IN SECONDS) OF DIFFERENT ALGORITHMS ON THE HOPKINS155 DATASET [54] WITH THE 12-D DATA POINTS OBTAINED USING PCA. SSRSC CAN ACHIEVE CLUSTERING ERROR OF 1.04% WHEN $s = 0.9$

Method	**SSC** [16]	**LRR** [18]	**LRSC** [29]	**LSR** [31]	**SMR** [32]	**S3C** [38]	**RSIM** [14]	**SSCOMP** [44]	**EnSC** [45]	**SSRSC**
Error (%)	2.18	3.28	5.41	2.33	2.27	2.61	1.76	5.35	1.81	**1.53**
Time (s)	0.49	0.28	**0.04**	0.05	0.35	2.86	0.18	0.06	0.05	0.07

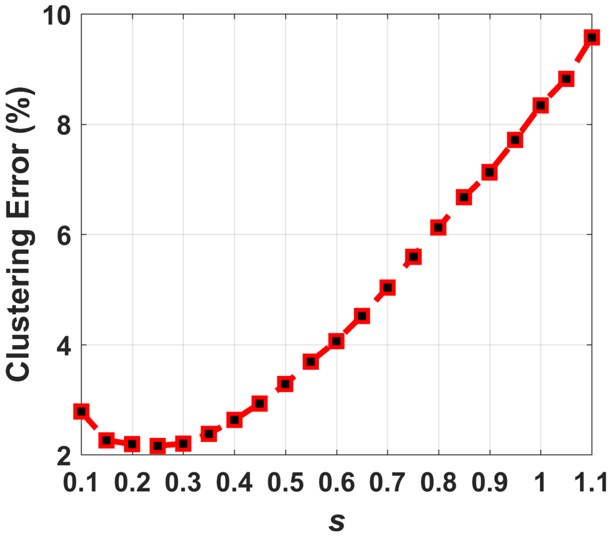

Fig. 8. Average clustering errors (%) of the proposed SSRSC algorithm with different scalar s on the Extended Yale B dataset [55].

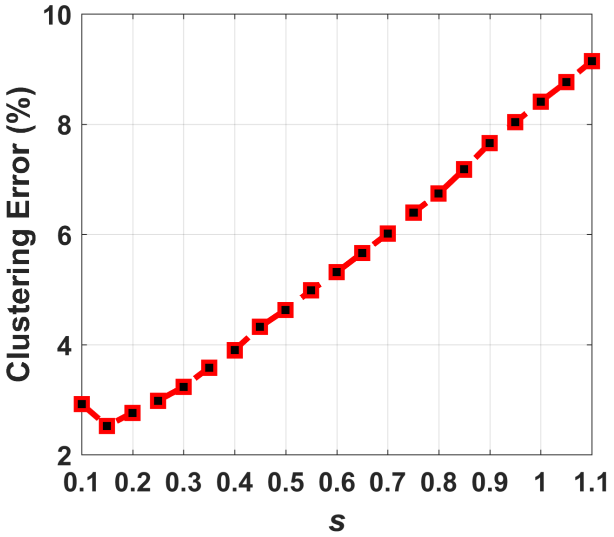

Fig. 9. Average clustering errors (%) of the proposed SSRSC with different s on MNIST [50], with 600 images for each digit.

SSC, 28.54% for LRR, 30.05% for LRSC, 28.59% for LSR, 28.18% for SMR, 5.16% for S3C, 6.56% for RSIM, 14.80% for SSCOMP, and 5.96% for EnSC. The performance gain of SSRSC (5.10%) over its baseline LSR (28.59%) demonstrates the power of simplex representation on human face clustering. In terms of running time, the proposed SSRSC algorithm is shown to be comparable to SSCOMP, which is the most efficient algorithm among all competing methods. In Table III, we can observe that the proposed SSRSC method achieves a lower clustering error than the previous methods except the deep learning-based DSC. This demonstrates the advantages of the proposed SSR representation over previous sparse or low-rank representation frameworks on human face clustering.

Results on Handwritten Digit Clustering: For the digit clustering problem, we evaluated the proposed SSRSC with the competing methods on the MNIST dataset [50] and the EMNIST dataset [85]. We follow the experimental settings in SSCOMP [44]. On the MNIST [50], the testing data consist of a randomly chosen number of $N_i \in \{50, 100, 200, 400, 600\}$ images for each of the 10 digits, while on the EMNIST [85], the testing data consist of $N_i = 500$ images for each of the 26 digits or letters. The feature vectors are extracted from the SCN [87]. The features extracted from the digit/letter images are originally 3472-D, and are projected onto a 500-D subspace using PCA.

We evaluate the influence of the scalar s on the average clustering error (%) of SSRSC on the MNIST dataset. The curve of clustering errors with respect to s is plotted in Fig. 9. Average clustering errors are computed over 20 trials sampled from 6000 randomly selected images ($N_i = 600$). As can be seen, the proposed SSRSC algorithm achieves an average clustering error of 4.53% and obtains its lowest average clustering error (2.52%) when $s = 0.15$, with 600 images for each digit from the MNIST dataset. Similar results can be observed on the EMNIST dataset [85]. The parameter λ is set as $\lambda = 0.01$.

We list the average clustering errors (%) of the comparison methods in Tables IV and V. It can be seen that the proposed SSRSC algorithm performs better than all other competing methods. For example, on the MNIST dataset, SSRSC achieves clustering errors of 11.81%, 6.90%, 5.65%, 5.31%, and 4.53% when the number of digit images are 500, 1000, 2000, 4000, and 6000, respectively. On the EMNIST dataset, SSRSC achieves clustering error of 23.45%, when the number of images for each digit/letter are 500. The results of the proposed SSRSC on the two datasets are much better than the other competing algorithms, including the recently proposed EnSC and ESC, respectively. This validates the advantages of the proposed SSR over previous sparse or LRR, for handwritten digit/letter clustering.

D. Ablation Study: Effectiveness of Simplex Representation

We perform comprehensive ablation studies on the Hopkins155 [54], Extended Yale B [55], and MNIST [50] datasets to validate the effectiveness of the proposed SSRSC model. SSRSC includes two constraints, that is, $\boldsymbol{c} \geq 0$ and $\mathbf{1}^\top \boldsymbol{c} = s$. To analyze the effectiveness of each constraint, we compare the proposed SSRSC model with several baseline methods.

TABLE II
AVERAGE CLUSTERING ERRORS (%) AND SPEED (IN SECONDS) OF DIFFERENT ALGORITHMS ON THE EXTENDED YALE B DATASET [55] WITH THE 6n-DIMENSIONAL (n IS THE NUMBER OF SUBJECTS) DATA POINTS OBTAINED USING PCA

# of Subjects	2 Subjects		3 Subjects		5 Subjects		8 Subjects		10 Subjects	
	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)
SSC [16]	1.86	0.17	3.10	0.31	4.31	0.74	5.85	2.72	10.94	4.27
LRR [18]	4.99	0.15	6.86	0.29	14.04	0.65	24.18	1.30	28.54	1.81
LRSC [29]	4.42	**0.03**	6.14	0.06	13.06	0.13	24.12	0.28	30.05	0.45
LSR [31]	4.98	**0.03**	6.87	0.05	14.14	0.12	24.39	0.25	28.59	0.36
SMR [32]	4.68	0.21	6.56	0.30	13.78	0.76	24.25	2.00	28.18	2.59
S3C [38]	1.27	1.31	2.71	2.58	3.41	3.52	4.13	7.14	5.16	13.98
RSIM [14]	2.17	0.09	2.96	0.17	4.13	0.47	5.82	1.77	6.56	3.19
SSCOMP [44]	2.76	**0.03**	5.35	**0.04**	7.89	**0.08**	11.40	0.18	14.80	**0.27**
EnSC [45]	1.64	**0.03**	2.91	0.05	3.84	0.11	5.84	0.20	5.96	0.29
SSRSC	1.26	**0.03**	1.58	**0.04**	2.70	**0.08**	5.66	**0.17**	5.10	**0.27**
SSRSC ($s = 0.25$)	**0.96**	**0.03**	**1.32**	**0.04**	**2.22**	**0.08**	**3.00**	**0.17**	**3.18**	**0.27**

TABLE III
AVERAGE CLUSTERING ERRORS (%) AND SPEED (IN SECONDS) OF DIFFERENT ALGORITHMS ON THE ORL DATASET [84] WITH THE 32 × 32 DATA POINTS OBTAINED BY RESIZING. SSRSC CAN ACHIEVE A CLUSTERING ERROR OF 21.25% WHEN $s = 0.4$

Method	**SSC** [16]	**LRR** [18]	**LRSC** [29]	**LSR** [31]	**SMR** [32]	**RSIM** [14]	**SSCOMP** [44]	**EnSC** [45]	**DSC** [41]	**SSRSC**
Error (%)	32.50	38.25	32.50	27.25	25.75	26.25	36.00	23.25	**14.00**	21.75
Time (s)	12.32	6.01	1.43	1.21	7.97	9.81	0.77	0.82	**0.23**	0.89

TABLE IV
AVERAGE CLUSTERING ERRORS (%) AND SPEED (IN SECONDS) OF DIFFERENT ALGORITHMS ON THE MNIST DATASET [50]. THE FEATURES ARE EXTRACTED FROM A SCATTERING NETWORK AND PROJECTED ONTO A 500-D SUBSPACE USING PCA. THE EXPERIMENTS ARE REPEATED 20 TIMES AND THE AVERAGE RESULTS ARE REPORTED

# of Images	500		1,000		2,000		4,000		6,000	
	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)	Error (%)	Time (s)
SSC [16]	16.99	27.36	15.95	49.25	14.42	94.84	14.00	239.95	14.40	423.47
LRR [18]	17.97	15.64	16.88	28.14	16.63	54.19	15.42	137.11	15.45	241.98
LRSC [29]	24.16	0.34	21.58	0.96	21.91	4.74	20.94	27.40	20.09	78.21
LSR [31]	24.98	**0.29**	20.24	0.91	20.56	4.74	22.24	28.86	20.12	83.14
SMR [32]	18.45	19.56	13.97	35.18	9.46	67.75	9.14	171.39	7.36	302.48
S3C [38]	15.92	135.20	12.23	263.33	10.54	521.93	9.46	1308.50	9.34	2544.41
RSIM [14]	18.13	10.06	15.70	18.09	11.48	34.84	10.53	88.14	10.21	155.56
SSCOMP [44]	16.36	0.32	13.33	**0.88**	9.40	**4.50**	8.78	**26.94**	8.75	**76.84**
EnSC [45]	13.45	0.37	9.31	0.98	7.69	4.67	6.71	27.14	5.86	77.24
SSRSC	11.81	0.34	6.90	0.98	5.65	5.01	5.31	30.06	4.53	85.68
SSRSC ($s = 0.15$)	**10.29**	0.35	**5.40**	0.99	**4.36**	5.03	**3.09**	30.11	**2.52**	85.74

TABLE V
AVERAGE CLUSTERING ERRORS (%) AND SPEED (IN SECONDS) OF DIFFERENT ALGORITHMS ON THE EMNIST DATASET [85]. THE FEATURES ARE EXTRACTED FROM A SCATTERING NETWORK AND PROJECTED ONTO A 500-D SUBSPACE USING PCA. THE EXPERIMENTS ARE REPEATED 10 TIMES AND THE AVERAGE RESULTS ARE REPORTED. SSRSC CAN ACHIEVE CLUSTERING ERROR OF 22.76% WHEN $s = 0.35$

Method	**SSC** [16]	**LRR** [18]	**LRSC** [29]	**LSR** [31]	**SMR** [32]	**RSIM** [14]	**SSCOMP** [44]	**EnSC** [45]	**ESC** [47]	**SSRSC**
Error (%)	30.11	33.89	31.51	32.21	29.58	30.15	35.66	26.23	25.42	**23.45**
Time (s)	863.29	495.13	162.32	172.28	621.89	320.14	159.22	160.98	**157.67**	186.23

TABLE VI
SUMMARY OF THE PROPOSED SSRSC AND THREE BASELINE METHODS LSR, NLSR, AND SLSR

Model	Data Term	Regularization Term	Constraint		Solution
			$\boldsymbol{C} \geq 0$	$\mathbf{1}^\top \boldsymbol{C} = s\mathbf{1}^\top$	
LSR (20)	$\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2$	$\|\boldsymbol{C}\|_F^2$	✗	✗	$\overline{\boldsymbol{C}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{X}$
NLSR (21)			✓	✗	See Supplementary File
SLSR (22)			✗	✓	See Supplementary File
SSRSC (9)			✓	✓	See §III-B

The first baseline is the trivial LSR model as follows:

$$\text{LSR: } \min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2. \tag{20}$$

By removing each individual constraint in the simplex constraint of the proposed SSRSC model (9), we have the second baseline method, called the non-negative LSR (NLSR) model

$$\text{NLSR: } \min_{\boldsymbol{C}} \quad \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2 \quad \text{s.t.} \quad \boldsymbol{C} \geq 0. \tag{21}$$

The NLSR model can be formulated by either removing the scaled affinity constraint $\mathbf{1}^\top\boldsymbol{c} = s$ from SSRSC or adding a non-negative constraint to the LSR model (20). For structural

TABLE VII

AVERAGE CLUSTERING ERRORS (%) OF LSR, NLSR, SLSR ($s = 0.9$), AND SSRSC ($s = 0.9$) ON HOPKINS155 [54], WITH THE 12-D DATA POINTS OBTAINED BY PCA

Method	LSR	NLSR	SLSR	SSRSC
Error (%)	3.67	1.75	3.16	**1.04**

TABLE VIII

AVERAGE CLUSTERING ERRORS (%) OF LSR, NLSR, SLSR ($s = 0.22$), AND SSRSC ($s = 0.25$) ON EXTENDED YALE B [55], WITH THE $6n$-D (n IS THE NUMBER OF SUBJECTS) DATA POINTS OBTAINED BY PCA

n	LSR	NLSR	SLSR	SSRSC
2	4.98	3.46	4.35	**0.96**
3	6.87	4.95	6.56	**1.32**
5	14.14	10.03	13.77	**2.22**
8	24.39	17.74	22.14	**3.00**
10	28.59	20.29	26.31	**3.18**

clearance of this article, we put the solution of the NLSR model (21) in the supplementary file. We can also remove the non-negative constraint $\boldsymbol{C} \geq 0$ in the scaled simplex set $\{\boldsymbol{C} \in \mathbb{R}^{N\times N} | \boldsymbol{C} \geq 0, \mathbf{1}^{\top}\boldsymbol{C} = s\mathbf{1}^{\top}\}$, and obtain a scaled-affine LSR (SLSR) model

$$\text{SLSR: } \min_{\boldsymbol{C}} \quad \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 + \lambda\|\boldsymbol{C}\|_F^2 \quad \text{s.t.} \quad \mathbf{1}^{\top}\boldsymbol{C} = s\mathbf{1}^{\top}. \tag{22}$$

The solution of the SLSR model (22) is also provided in the supplementary file.

The proposed SSRSC method, as well as the three baseline methods, can be expressed in a standard form

$$\begin{aligned} \min_{\boldsymbol{C}} \quad & \textit{Data Term} + \textit{Regularization Term} \\ \text{s.t.} \quad & \textit{Constraints}. \end{aligned} \tag{23}$$

For the NLSR, SLSR, and SSRSC methods, the data and regularization terms are the same as those of the basic LSR model (9), that is, $\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2$ and $\lambda\|\boldsymbol{C}\|_F^2$, respectively. The difference among these comparison methods lies in the *Constraints*. In Table VI, we summarize the proposed SSRSC, and the three baseline methods LSR, NLSR, and SLSR.

In Tables VII–IX, we list the average clustering errors (%) of the proposed SSRSC and three baseline methods on the Hopkins155 [54], Extended Yale B [55], and MNIST [50] datasets. Note that here we tune the parameter s to achieve the best performance of SSRSC on different datasets.

Effectiveness of the Non-Negative Constraint: The effectiveness of the non-negative constraint $\boldsymbol{C} \geq 0$ in the proposed SSRSC model can be validated from two complementary aspects. First, it can be validated by evaluating the performance improvement of the baseline methods NLSR (21) over LSR [31]. Since the only difference between them is the additional non-negative constraint in NLSR (21), the performance gain of NSLR over LSR can directly reflect the effectiveness of the non-negative constraint. Second, the effectiveness of the non-negative constraint can also be validated by comparing the performance of the proposed SSRSC (9) and the baseline method SLSR (22). Since SLSR (22) lacks a non-negative constraint when compared to SSRSC, the performance gain of SSRSC over SLSR should be due to the introduced non-negativity.

TABLE IX

AVERAGE CLUSTERING ERRORS (%) OF LSR, NLSR, SLSR ($s = 0.24$), AND SSRSC ($s = 0.15$) ON THE MNIST DATASET [50]. THE FEATURES OF THE DATA POINT ARE EXTRACTED FROM A SCATTERING NETWORK [87] AND PROJECTED ONTO A 500-D SUBSPACE OBTAINED USING PCA. THE EXPERIMENTS ARE INDEPENDENTLY REPEATED 20 TIMES

# of Images	LSR	NLSR	SLSR	SSRSC
500	24.98	15.29	22.00	**10.29**
1,000	20.24	12.66	18.59	**5.40**
2,000	20.56	10.13	16.87	**4.36**
4,000	22.24	9.07	15.48	**3.09**
6,000	20.12	8.34	14.23	**2.52**

The results listed in Table VII show that on the Hopkins155 dataset, the baseline methods LSR and NLSR achieve average clustering errors of 3.67% and 1.75%, respectively. This demonstrates that by adding the non-negative constraint to LSR, the resulting baseline method NLSR has a significantly reduced clustering error. Besides, the average clustering errors of SSRSC ($s = 0.9$) and SLSR ($s = 0.9$) are 1.04% and 3.16%, respectively. This shows that if we remove the non-negative constraint from SSRSC, the resulting SLSR will have a significantly lower clustering performance. Similar trends can be found from Tables VIII and IX on the Extended Yale B dataset [55] and the MNIST dataset [50], respectively. These results validate the contribution of the non-negative constraint to the success of the proposed SSRSC model.

Effectiveness of the Scaled-Affine Constraint: The effectiveness of the scaled-affine constraint can be validated from two aspects. First, it can be validated by comparing the baseline methods LSR (20) and SLSR (22), which are summarized in Table VI. Since the only difference between them is that SLSR contains an additional scaled affine constraint over LSR, the performance gain of SLSR over LSR can directly validate the scaled affine constraint's effectiveness. Second, the effectiveness can also be validated by comparing the proposed SSRSC (9) and the baseline NLSR (21). This is because NLSR removes the scaled affine constraint from SSRSC.

From the results listed in Tables VII, we observe that on the Hopkins155 dataset [54], the proposed SSRSC ($s = 0.9$) can achieve lower clustering error (1.04%) than the baseline method NLSR (1.75%). Similar conclusions can be drawn from Tables VIII and IX for experiments on the Extended Yale B dataset [55] and the MNIST dataset [50], respectively. All of these comparisons demonstrate that the scaled affine constraint is another essential factor for the success of SSRSC.

Effectiveness of the Simplex Constraint: We find that the proposed simplex constraint, that is, the integration of the non-negative and scaled affine constraints, can further boost the performance of SC. This can be validated by comparing the performance of the proposed SSRSC (9) and the baseline method LSR (20). The results listed in Tables VII–IX show that on all three commonly used datasets, the proposed SSRSC can achieve much lower clustering errors than the baseline method LSR. For example, on the MNIST dataset, the clustering errors of SSRSC ($s = 0.15$) are 10.29%, 5.40%, 4.36%, 3.09%, and 2.52% when we randomly select

TABLE X
AVERAGE CLUSTERING ERRORS (%) OF SSRSC ($s = 0.9$), SSRSC-DIAG ($s = 0.8$), AND RGC ON THE HOPKINS155 DATASET [54], WITH THE 12-D DATA POINTS OBTAINED USING PCA

Method	**SSRSC**	**SSRSC-diag**	**RGC** [88]
Error (%)	**1.04**	1.87	1.76

50, 100, 200, 400, and 600 images for each of the 10 digits, respectively. Meanwhile, the corresponding clustering errors of the baseline method LSR are 24.98%, 20.24%, 20.56%, 22.24%, 20.12%, respectively. SSRSC performs much better than LSR in all cases. Similar trends can also be found in Table VII for the Hopkins155 dataset and in Table VIII for the Extended Yale B dataset.

Influence of Diagonal Constraint: One key difference of our SSRSC model to the previous models is that in SSRSC, we get rid of the diagonal constraint $\text{diag}(\boldsymbol{C}) = \boldsymbol{0}$ considering that the practical data are often noisy. To achieve this, we compare the method on the Hopkins155 dataset, together with the SSRSC model with an additional diagonal constraint of $\text{diag}(\boldsymbol{C}) = \boldsymbol{0}$ (we call this method SSRSC-diag). The results are listed in Table X. One can see that the variant SSRSC-diag achieves inferior performance with the original SSRSC. We also compare a state-of-the-art clustering method, that is, RGC [88], designed specifically for clustering noisy data. RGC achieves slightly better performance than SSRSC-diag, but is still inferior to our SSRSC without the diagonal constraint. This demonstrates the necessity of getting rid of the diagonal constraint in our SSRSC model to deal with noisy data.

V. CONCLUSION

In this article, we proposed an SSR-based model for spectral clustering-based SC. Specifically, we introduced the nonnegative and scaled affine constraints into a simple LSR model. The proposed SSRSC model can reveal the inherent correlations among data points in a highly discriminative manner. Based on the SSRSC model, a novel spectral clustering-based algorithm was developed for SC. Extensive experiments on three benchmark clustering datasets demonstrated that the proposed SSRSC algorithm is very efficient and achieves better clustering performance than the state-of-the-art SC algorithms. The significant improvements of SSRSC over the baseline models demonstrated the effectiveness of the proposed simplex representation.

This article can be extended in at least three directions. First, it is promising to extend the proposed SSRSC model to the nonlinear version by using kernel approaches [60], [69], [70], [89]. Second, it is worth accelerating the proposed algorithm for scalable SC [47], [90]. Third, adapting the proposed SSRSC model for imbalanced datasets [47], [89] is also a valuable direction.

REFERENCES

[1] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[2] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. J. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. CVPR*, 2003, pp. 11–18.

[3] M. C. Tsakiris and R. Vidal, "Algebraic clustering of affine subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 482–489, Feb. 2018.

[4] J. P. Costeira and T. Kanade, "A multibody factorization method for independent moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[5] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1–15, Dec. 2005.

[6] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[7] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the EM algorithm," in *Proc. CVPR*, 2004, pp. 707–714.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1832–1845, Oct. 2010.

[10] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Proc. ECCV*, 2006, pp. 94–106.

[11] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," in *Proc. CVPR*, 2010, pp. 217–240.

[12] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Proc. CVPR*, 2007, pp. 1–6.

[13] G. Chen and G. Lerman, "Spectral curvature clustering," *Int. J. Comput. Vis.*, vol. 81, no. 3, pp. 317–330, 2009.

[14] P. Ji, M. Salzmann, and H. Li, "Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data," in *Proc. ICCV*, 2015, pp. 4687–4695.

[15] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE CVPR*, 2009, pp. 2790–2797.

[16] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[17] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. ICML*, 2010, pp. 663–670.

[18] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[19] J. Chen and J. Yang, "Robust subspace segmentation via low-rank representation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1432–1445, Aug. 2014.

[20] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Robust semi-supervised subspace clustering via non-negative low-rank representation," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1828–1838, Aug. 2016.

[21] J. Wang, X. Wang, F. Tian, C. H. Liu, and H. Yu, "Constrained low-rank representation for robust subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4534–4546, Dec. 2017.

[22] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[23] Z. Wen, B. Hou, Q. Wu, and L. Jiao, "Discriminative transformation learning for fuzzy sparse subspace clustering," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2218–2231, Aug. 2018.

[24] J. Wen, Y. Xu, and H. Liu, "Incomplete multiview spectral clustering with adaptive graph learning," *IEEE Trans. Cybern.*, to be published.

[25] J. Lee, H. Lee, M. Lee, and N. Kwak, "Nonparametric estimation of probabilistic membership for subspace clustering," *IEEE Trans. Cybern.*, to be published.

[26] M. Brbić and I. Kopriva, "ℓ_0-motivated low-rank sparse subspace clustering," *IEEE Trans. Cybern.*, to be published.

[27] H. Zhang, J. Yang, F. Shang, C. Gong, and Z. Zhang, "LRR for subspace segmentation via tractable Schatten-p norm minimization and factorization," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1722–1734, May 2019.

[28] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo, "Dual graph regularized latent low-rank representation for subspace clustering," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4918–4933, Dec. 2015.

[29] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. CVPR*, 2011, pp. 1801–1807.

[30] V. Karavasilis, K. Blekas, and C. Nikou, "A novel framework for motion segmentation and tracking by clustering incomplete trajectories," *Comput. Vis. Image Understanding*, vol. 116, no. 11, pp. 1135–1148, 2012.

[31] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. ECCV*, 2012, pp. 347–360.

[32] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *Proc. CVPR*, 2014, pp. 3834–3841.

[33] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. CVPR*, Jun. 2013, pp. 430–437.

[34] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proc. CVPR*, 2014, pp. 3818–3825.

[35] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Ordered subspace clustering with block-diagonal priors," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3209–3219, Dec. 2016.

[36] B. Li, Y. Zhang, Z. Lin, and H. Lu, "Subspace clustering by mixture of Gaussian regression," in *Proc. CVPR*, 2015, pp. 2094–2102.

[37] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *Proc. CVPR*, 2015, pp. 277–286.

[38] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Trans. Image Process.*, vol. 26, no. 6. pp. 2988–3001, Jun. 2017.

[39] J. Xu, K. Xu, K. Chen, and J. Ruan, "Reweighted sparse subspace clustering," *Comput. Vis. Image Understanding*, vol. 138, pp. 25–37, Sep. 2015.

[40] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *Proc. CVPR*, 2017, pp. 682–691.

[41] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. D. Reid, "Deep subspace clustering networks," in *Proc. NIPS*, 2017, pp. 24–33.

[42] M. Yin, S. Xie, Z. Wu, Y. Zhang, and J. Gao, "Subspace clustering via learning an adaptive low-rank graph," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3716–3728, Aug. 2018.

[43] J. Yang, J. Liang, K. Wang, Y.-L. Yang, and M.-M. Cheng, "Automatic model selection in subspace clustering via triplet relationships," in *Proc. AAAI*, 2018, pp. 4358–4365.

[44] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. CVPR*, 2016, pp. 3918–3927.

[45] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. CVPR*, 2016, pp. 3928–3937.

[46] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *Proc. CVPR*, 2017, pp. 4323–4332.

[47] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *Proc. ECCV*, Sep. 2018, pp. 1–17.

[48] U. Von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

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

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

[51] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B (Meth.)*, vol. 58, no. 1, pp. 267–288, 1996.

[52] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *Proc. IEEE CVPR*, 2012, pp. 2328–2335.

[53] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[54] R. Tron and R. Vidal, "A benchmark for the comparison of 3D motion segmentation algorithms," in *Proc. CVPR*, 2007, pp. 1–8.

[55] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[56] A.-L. Ellis and J. Ferryman, "Biologically-inspired robust motion segmentation using mutual information," *Comput. Vis. Image Understand.*, vol. 122, pp. 47–64, May 2014.

[57] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[58] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[59] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[60] C. Peng, Z. Kang, and Q. Cheng, "Integrating feature and graph learning with low-rank representation," *Neurocomputing*, vol. 249, pp. 106–116, Aug. 2017.

[61] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. AAAI*, 2015, pp. 3827–3833.

[62] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 244–252.

[63] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proc. ICCV*, 2017, pp. 1105–1113.

[64] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2996–3010, Jun. 2018.

[65] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang, "A hybrid L1–L0 layer decomposition model for tone mapping," in *Proc. CVPR*, Jun. 2018, pp. 4758–4766.

[66] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proc. ECCV*, 2018, pp. 21–38.

[67] C. Peng, Z. Kang, H. Li, and Q. Cheng, "Subspace clustering using log-determinant rank approximation," in *Proc. ACM SIGKDD*, 2015, pp. 925–934.

[68] L. Zhuang *et al.*, "Constructing a nonnegative low-rank and sparse graph with data-adaptive features," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3717–3728, Nov. 2015.

[69] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, Dec. 2017.

[70] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl. Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.

[71] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu, "Clustering with similarity preserving," *Neurocomputing*, vol. 365, pp. 211–218, Nov. 2019.

[72] J. Xu, W. An, L. Zhang, and D. Zhang, "Sparse, collaborative, or nonnegative representation: Which helps pattern classification?" *Pattern Recognit.*, vol. 88, pp. 679–688, Apr. 2019.

[73] Q. Li, W. Liu, and L. Li, "Affinity learning via a diffusion process for subspace clustering," *Pattern Recognit.*, vol. 84, pp. 39–50, Dec. 2018.

[74] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering," in *Proc. IJCAI*, 2015, pp. 3569–3575.

[75] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bull. Amer. Math. Soc.*, vol. 49, no. 1, pp. 1–23, 1943.

[76] J. Eckstein and D. P. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, nos. 1–3, pp. 293–318, 1992.

[77] K. S. Riedel, "A Sherman–Morrison–Woodbury identity for rank augmenting matrices with application to centering," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 2, pp. 659–662, 1992.

[78] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.

[79] E. Elhamifar, G. Sapiro, and S. Sastry, "Dissimilarity-based sparse subset selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2182–2197, Nov. 2016.

[80] C. Michelot, "A finite algorithm for finding the projection of a point onto the canonical simplex of RN," *J. Optim. Theory Appl.*, vol. 50, no. 1, pp. 195–200, Jul. 1986.

[81] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the L1-ball for learning in high dimensions," in *Proc. ACM ICML*, 2008, pp. 272–279.

[82] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Math. Program.*, vol. 158, nos. 1–2, pp. 575–585, Jul. 2016.

[83] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering analysis and an algorithm," in *Proc. NIPS*, vol. 14, 2001, pp. 849–856.

[84] F. S. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 1994, pp. 138–142.

[85] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 2921–2926.

[86] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal Component Analysis*. New York, NY, USA: Springer, 1986, pp. 115–128.

[87] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[88] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, to be published.

[89] C. Peng and Q. Cheng, "Discriminative regression machine: A classifier for high-dimensional data or imbalanced data," *arXiv 1904.07496*, 2019.

[90] C. Peng, C. Chen, Z. Kang, J. Li, and Q. Cheng, "RES-PCA: A scalable approach to recovering low-rank matrices," in *Proc. CVPR*, Jun. 2019, pp. 7317–7325.