

Scaled Simplex Representation for Subspace Clustering

Jun Xu^{1,2}, Mengyang Yu², Ling Shao², *Senior Member, IEEE*, Wangmeng Zuo³, *Senior Member, IEEE*,
Deyu Meng⁴, Lei Zhang⁵, *Fellow, IEEE*, David Zhang^{5,6}, *Fellow, IEEE*

¹College of Computer Science, Nankai University, Tianjin, China

²Inception Institute of Artificial Intelligence, Abu Dhabi, UAE

³School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

⁴School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

⁵Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China

⁶School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China

The self-expressive property of data points, i.e., each data point can be linearly represented by the other data points in the same subspace, has proven effective in leading subspace clustering 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 paper, we introduce a scaled simplex representation (SSR) for subspace clustering 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 subspace clustering (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 state-of-the-art subspace clustering methods on accuracy. The code can be found at <https://github.com/csjunxu/SSRSC>.

Index Terms—Subspace clustering, scaled simplex representation, self-expressiveness.

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 subsequent algorithms. To this end, many image processing tasks require to cluster 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: iterative methods [2], [3], algebraic methods [4], [5], [14], statistical methods [6]–[9], and self-expressive methods [10]–[13], [15]–[47]. Among these methods, self-expressive ones are 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], and 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 real-world cases where the 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 subspace clustering. 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. Introducing the non-negative constraint has three primary benefits: 1) in **Step 1**, it eliminates the physically problematic subtractions among

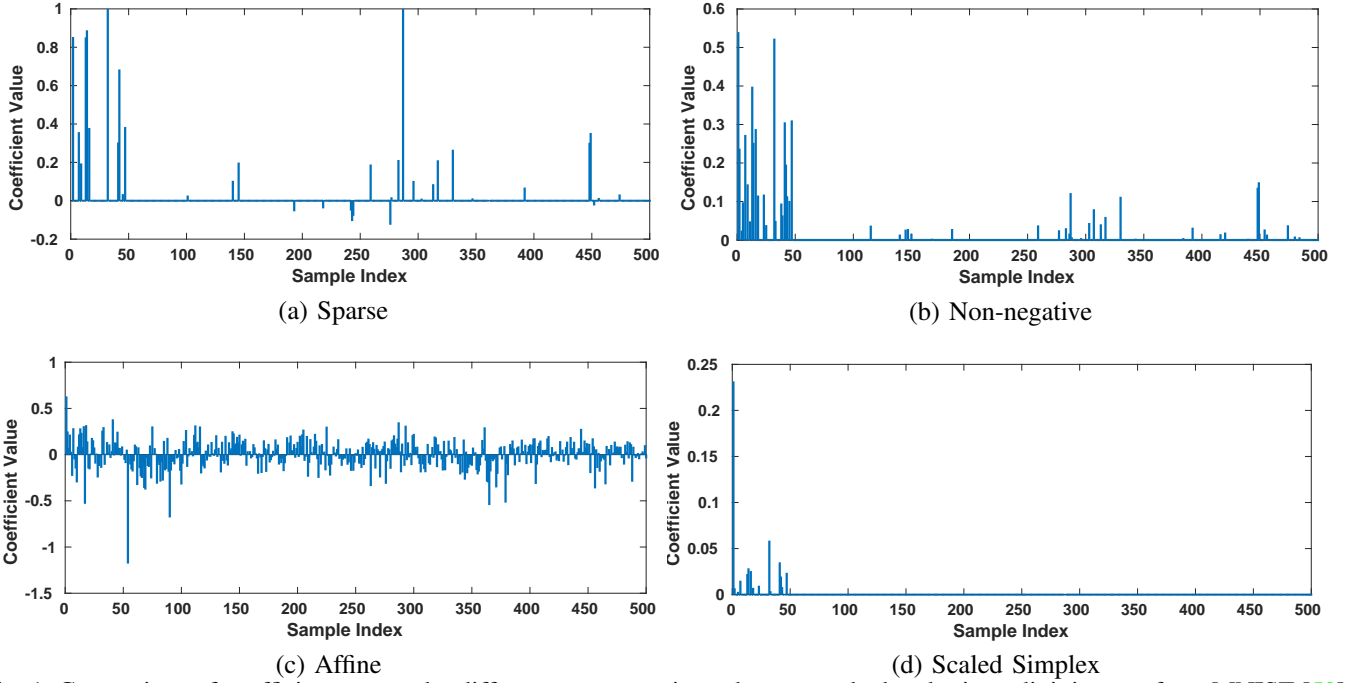


Fig. 1: Comparison of coefficient vectors by different representation schemes on the handwritten digit images from MNIST [50]. A digit sample “0” is used to compute the vector over 500 samples of digits $\{0, 1, \dots, 9\}$ (50 for each). (a) The vector solved by a sparse model (e.g., LASSO [51]) are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

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 scaled simplex representation, in Figure 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 $\{0, 1, \dots, 9\}$ (50 for each). We observe that: The vector solved by a sparse model (e.g., LASSO [51]) are confusing, the coefficients over the samples of other digits are also non-zero (Figure 1 (a)). The vector solved by the least square regression (LSR) model with non-negative constraint are noisy (Figure 1 (b)). The coefficients solved by LSR with affine constraint are densely distributed over all samples (Figure 1 (c)). The LSR with the proposed scaled simplex representation (SSR) can obtain physically more meaningful coefficients vector (Figure 1 (d)).

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

- We introduce a new Scaled Simplex Representation (SSR) for subspace clustering. The proposed SSR can not

only maintain the inherent correlations among the data points, but also handle practical problems more flexibly.

- We propose an SSRSC model for subspace clustering. 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 (ADMM) [53]. Each variable can be updated efficiently, and the convergence can be guaranteed.
- We performed comprehensive experiments on several benchmark datasets, i.e., 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 state-of-the-art SC algorithms on motion segmentation, face clustering, and hand-written digits/letters clustering.

The remainder of this paper is organized as follows. In §II, we briefly survey the related work. In §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 §IV to evaluate the SSRSC algorithm and compare it with state-of-the-art SC algorithms. Conclusion and future work are given in §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: iterative

methods, algebraic methods, statistical methods, and 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 our work. They usually use local information around each data point to compare the similarity of two points. Inspired by compressed sensing theory [57], [58], Sparse Subspace Clustering (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 a low-rank representation of the data over themselves. Lu *et al.* proposed a clustering method based on Least Squares Regression (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 a state-of-the-art performance for the subspace clustering 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 subspace clustering (EnSC) model [45] which correctly predicted relevant connections among different clusters. Ji *et al.* [41] developed the first unsupervised network for subspace clustering by learning the *self-expressiveness* property [15].

B. The Self-expressiveness Based Framework

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

$$\mathbf{x}_i = \mathbf{X} \mathbf{c}_i, \quad (1)$$

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

$$\mathbf{X} = \mathbf{X} \mathbf{C}, \quad (2)$$

where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the coefficient matrix. To find the desired \mathbf{C} , 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:

Sparse Subspace Clustering (SSC) [16]:

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

Low-Rank Representation (LRR) [18]:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X} \mathbf{C}\|_{2,1} + \lambda \|\mathbf{C}\|_*. \quad (4)$$

Least Squares Regression (LSR) [31]:

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

SSC by Orthogonal Matching Pursuit (SSCOMP) [44]:

$$\min_{\mathbf{c}_i} \|\mathbf{x}_i - \mathbf{X} \mathbf{c}_i\|_2^2 \quad \text{s.t.} \quad \|\mathbf{c}_i\|_0 \leq p, c_{ii} = 0. \quad (6)$$

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

$$\mathbf{A} = (|\mathbf{C}| + |\mathbf{C}^\top|)/2. \quad (7)$$

After the affinity matrix \mathbf{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 \mathbf{C} to be positive in \mathbf{A} , and hence damage the inherent correlations among the data points in \mathbf{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 paper, we introduce the Scaled Simplex Representation 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 subspace clustering and other vision tasks [62]–[66], there are other constraints also employed by existing subspace clustering algorithms. In [34], Feng *et al.* proposed the Laplacian constraint for block-diagonal matrix pursuit. The developed block-diagonal SSC and LRR show clear improvements over SSC and LRR on subspace clustering and graph construction for semi-supervised 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 work for similarity graph learning [68]–[72]. In these work, the non-negativity is employed to construct sparse or low-rank similarity graph. Our work shares the same spirit with these work on this point, and aims to build physically reasonable affinity matrix by employing non-negativity. Recently, Li *et al.* [73] proposed to directly learn the affinity matrix by diffusion process to spread local manifold structure of data along with its global manifold. This work can be viewed as propagating data manifold constraint into sparsity models to enhance its connectivity for subspace clustering. The simplex sparse constraint in [74] is closely related to our work. But our proposed scaled simplex constraint allows the sum of coefficients to be a scalar, while in this work the sum is fixed to be 1. Besides, to make our model simpler, we do not use the diagonal constraint in [74].

Algorithm 1: Projection of vector \mathbf{u}_{k+1} onto a simplex

Input: Data point $\mathbf{u}_{k+1} \in \mathbb{R}^N$, scalar s ;
 1. Sort \mathbf{u}_{k+1} into \mathbf{w} : $w_1 \geq w_2 \geq \dots \geq w_N$;
 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: \mathbf{z}_{k+1} : $z_{k+1}^i = \max\{u_{k+1}^i + \beta, 0\}$, $i = 1, \dots, N$.

III. SIMPLEX REPRESENTATION BASED SUBSPACE CLUSTERING

In this section, we propose a Scaled Simplex Representation (SSR) based subspace clustering (SSRSC) model, develop an optimization algorithm to solve it, and present a novel SSRSC based algorithm for subspace clustering (SC).

A. Proposed SSRSC Model

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

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

where $\mathbf{1}$ is the vector of all ones and $s > 0$ is a scalar denoting the sum of entries in the coefficient vector \mathbf{c}_i . We use the term ‘‘scaled simplex’’ here because the entries in the coefficient vector \mathbf{c}_i are constrained by a scaled simplex, i.e., 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:

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

where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the coefficient matrix. Here, the constraint $\mathbf{C} \geq 0$ 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 \mathbf{C} discriminative. The constraint $\mathbf{1}^\top \mathbf{C} = s\mathbf{1}^\top$ limits the sum of each coefficient vector \mathbf{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 \mathbf{Z} into the SSRSC model (9), and reformulate it as a linear equality-constrained problem:

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

whose solution w.r.t. \mathbf{C} coincides with the solution of (9). Since the objective function in Eqn. (10) is separable w.r.t. the

variables \mathbf{C} and \mathbf{Z} , it can be solved using the ADMM [53]. The corresponding augmented Lagrangian function is

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

where Δ is the augmented Lagrangian multiplier and $\rho > 0$ is the penalty parameter. Denote by $(\mathbf{C}_k, \mathbf{Z}_k)$ and Δ_k the optimization variables and Lagrange multiplier at iteration k ($k = 0, 1, 2, \dots$), respectively. We initialize the variables \mathbf{C}_0 , \mathbf{Z}_0 , and Δ_0 to be conformable zero matrices. By taking derivatives of Lagrangian function \mathcal{L} w.r.t. \mathbf{C} and \mathbf{Z} , and setting them to be zeros, we can alternatively update the variables as follows:

(1) **Updating \mathbf{C} while fixing \mathbf{Z}_k and Δ_k :**

$$\mathbf{C}_{k+1} = \arg \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \frac{\rho}{2} \|\mathbf{C} - (\mathbf{Z}_k + \frac{1}{\rho} \Delta_k)\|_F^2. \quad (12)$$

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

$$\mathbf{C}_{k+1} = (\mathbf{X}^\top \mathbf{X} + \frac{\rho}{2} \mathbf{I})^{-1} (\mathbf{X}^\top \mathbf{X} + \frac{\rho}{2} \mathbf{Z}_k + \frac{1}{2} \Delta_k). \quad (13)$$

We note that the complexity for updating \mathbf{C} is $\mathcal{O}(N^3)$ when there are N data points in the data matrix \mathbf{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 Eqn. (13) as

$$(\mathbf{X}^\top \mathbf{X} + \frac{\rho}{2} \mathbf{I})^{-1} = \frac{2}{\rho} \mathbf{I} - (\frac{2}{\rho})^2 \mathbf{X}^\top (\mathbf{I} + \frac{2}{\rho} \mathbf{X} \mathbf{X}^\top)^{-1} \mathbf{X}. \quad (14)$$

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

(2) **Updating \mathbf{Z} while fixing \mathbf{C}_k and Δ_k :**

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

This is a quadratic programming problem with 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 simplex. Denoting by \mathbf{u}_{k+1} an arbitrary column of $\frac{\rho}{2\lambda + \rho}(\mathbf{C}_{k+1} - \rho^{-1} \Delta_k)$, the solution of \mathbf{z}_{k+1} (the corresponding column in \mathbf{Z}_{k+1}) can be solved by projecting \mathbf{u}_{k+1} onto a simplex [81]. The solution of problem (15) is summarized in Algorithm 1.

Algorithm 2: Solve the SSRSC model (10) via ADMM

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

(3) Updating $\mathbf{\Delta}$ while fixing \mathbf{C}_k and \mathbf{Z}_k :

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

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, $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F \leq \text{Tol}$, $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F \leq \text{Tol}$, and $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F \leq \text{Tol}$ must be simultaneously satisfied, where $\text{Tol} = 0.01$ tolerates small errors. Since the objective function and constraints are both convex, the problem (10) solved by ADMM is guaranteed to converge at a global optimal solution. We summarize these updating procedures in Algorithm 2.

Convergence Analysis. The convergence of Algorithm 2 can be guaranteed since the overall objective function (10) is convex with a global optimal solution. In Figure 2, we plot the convergence curves of the errors of $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F$, $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F$, $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F$. One can see that they are reduced to less than $\text{Tol} = 0.01$ simultaneously in 5 iterations.

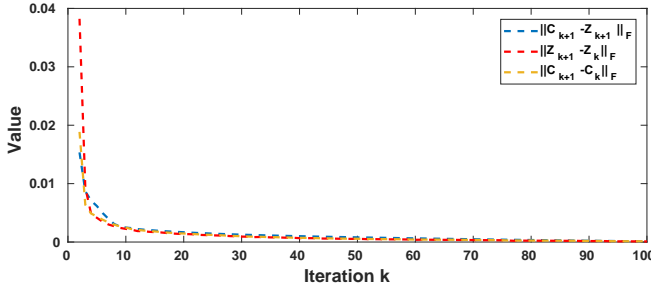


Fig. 2: The convergence curves of $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F$ (red line), $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F$ (blue line), and $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F$ (green line) of the proposed SSRSC on the “1R2RC” sequence from the Hopkins155 dataset [54].

C. Theoretical Analysis

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

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

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

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

$$f(\mathbf{C}^* + \lambda(\mathbf{D} - \mathbf{C}^*)) \geq f(\mathbf{C}^*). \quad (17)$$

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

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

Combining this with Eqn. (17), we obtain $\lambda f(\mathbf{D}) \geq \lambda f(\mathbf{C}^*)$ and hence $f(\mathbf{D}) \geq f(\mathbf{C}^*)$, $\forall \mathbf{D} \in \mathbb{R}^{N \times N}$. Then, \mathbf{C}^* is the minimum of f on hyper-plane $\bar{\mathcal{S}}$, resulting a contradiction. \square

D. Discussion

The constraint “ $\mathbf{1}^\top \mathbf{c} = 1$ ” has already been used in Sparse Subspace Clustering (SSC) [15], [16] to deal with the presence of affine, rather than linear, subspaces. However, limiting the sum of the coefficient vector \mathbf{c} to be 1 is not flexible enough for real-world clustering problems. What’s more, suppressing the sum of the entries in the coefficient vector \mathbf{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 §IV), we observe improved performance of SSRSC on subspace clustering 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 \mathbf{x}_i , indicating that each data point cannot be represented by itself, thus avoiding trivial solution of identity matrix. However, this brings additional computational costs and prevent the whole 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 trivial solution of identity matrix $\mathbf{C} = \mathbf{I}$. Second, $c_{ii} \neq 0$ has a clear physical meaning, allowing a sample \mathbf{x}_i in the subspace to be *partially* represented by itself. This is particularly useful when \mathbf{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 §IV-D.

Algorithm 3: Subspace Clustering by SSRSC

Input: A set of data points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ lying in a union of n subspaces $\{\mathcal{S}_j\}_{j=1}^n$;

1. Obtain the coefficient matrix \mathbf{C} by solving the SSRSC model:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_F^2 \text{ s.t. } \mathbf{C} \geq 0, \mathbf{1}^\top \mathbf{C} = \mathbf{s} \mathbf{1}^\top;$$

2. Construct the affinity matrix by

$$\mathbf{A} = \frac{\mathbf{C} + \mathbf{C}^\top}{2};$$

3. Apply spectral clustering [83] to the affinity matrix;

Output: Segmentation of data: $\mathbf{X}_1, \dots, \mathbf{X}_n$.

E. Subspace Clustering via Simplex Representation

Subspace Clustering Algorithm. Denote by $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^N$ 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 \mathbf{C} , and then construct a non-negative affinity matrix \mathbf{A} from \mathbf{C} by exponentiation [13], absolute symmetrization [15], [16], [29], [31]–[34], [36], [37], [39], [44], or 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

$$\mathbf{A} = (\mathbf{C} + \mathbf{C}^\top)/2. \quad (19)$$

As a common post-processing step in [15]–[18], [29], [31]–[34], [36], [37], [39], [40], [44], we apply the spectral clustering technique [83] to the affinity matrix \mathbf{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 subspace clustering algorithm is summarized in Algorithm 3.

Complexity Analysis. Assume that there are N data points in the data matrix \mathbf{X} . In Algorithm 2, the costs for updating \mathbf{C} and \mathbf{Z} are $\mathcal{O}(N^2D)$ and $\mathcal{O}(N^2 \log N)$, respectively. The costs for updating Δ and ρ are negligible compared to the updating costs of \mathbf{C} and \mathbf{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 $\mathbf{X} \in \mathbb{R}^{D \times N}$.

IV. EXPERIMENTS

In this section, we first compare the proposed SSRSC with state-of-the-art subspace clustering (SC) methods. The comparison are performed on five benchmark datasets on motion segmentation for video analysis, human faces clustering, and hand-written 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 five parameters to be determined in the ADMM algorithm: the regularization parameter λ , the sum



Fig. 3: Exemplar motion segmentation of one frame from different sequences in the Hopkins-155 dataset [54].

s of the entries in the coefficient vector, the penalty parameter ρ , and the iteration number K . In all the 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], the 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 §IV-C. All experiments are run under the Matlab2014b environment on a machine with a CPU of 3.50GHz and a 12GB RAM.

B. Datasets

We evaluate the proposed SSRSC method on the Hopkins-155 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 hand-written digits/letters clustering.

Hopkins-155 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 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-dimensional subspace, in which we evaluate the comparison methods. This dataset [54] has been widely used as a benchmark to evaluate SC methods for motion segmentation. Figure 3 presents some segmentation examples from the Hopkins-155 dataset [54], where different colors indicate different moving objects.

Extended Yale B dataset [55] contains face images of 38 human subjects, and each subject has 64 near-frontal images (gray-scale) 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



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

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 $n \in \{2, 3, 5, 8, 10\}$ 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. Figure 4 (top) shows some face images from Extended Yale B, captured under different lighting conditions.

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 Extended Yale B [55] for two reasons: 1) its varying face images varies much more complex; 2) it only contains 400 images, much smaller than Extended Yale B (2432 images). Figure 4 (down) shows some face images from ORL.

MNIST dataset [50] contains 60,000 gray-scale images of 10 digits (i.e., $\{0, \dots, 9\}$) 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 of 3,472-dimension. The feature vectors for all images are then projected onto a 500-dimensional subspace using PCA. Figure 5 shows some examples of the handwritten digit images in this dataset.

EMNIST dataset [85] is an extension of the MNIST dataset that contains gray-scale handwritten digits and letters. This dataset contains overall 190,998 images corresponding to 26 lower case 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 a scattering convolution network (SCN) [87], which is translation invariant and deformation stable. The feature vectors are of 3,472-



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

dimension, the feature vectors for all images are projected onto a 500-dimensional 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 ablation study, i.e., $s = 0.9$ on Hopkins-155, $s = 0.25$ 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 §II as described in [32], for fair comparison. For the other methods, we tune their corresponding parameters on each of the three datasets, i.e., the Hopkins-155 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 6,000 images for each digit in $\{0, 1, \dots, 9\}$. We randomly select 50 images for each digit, and use the 500 total images to construct the affinity matrices using these competing SC methods. The results are visualized in Figure 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 other compared methods.

Results on Motion Segmentation. We first study how the 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 Figure 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

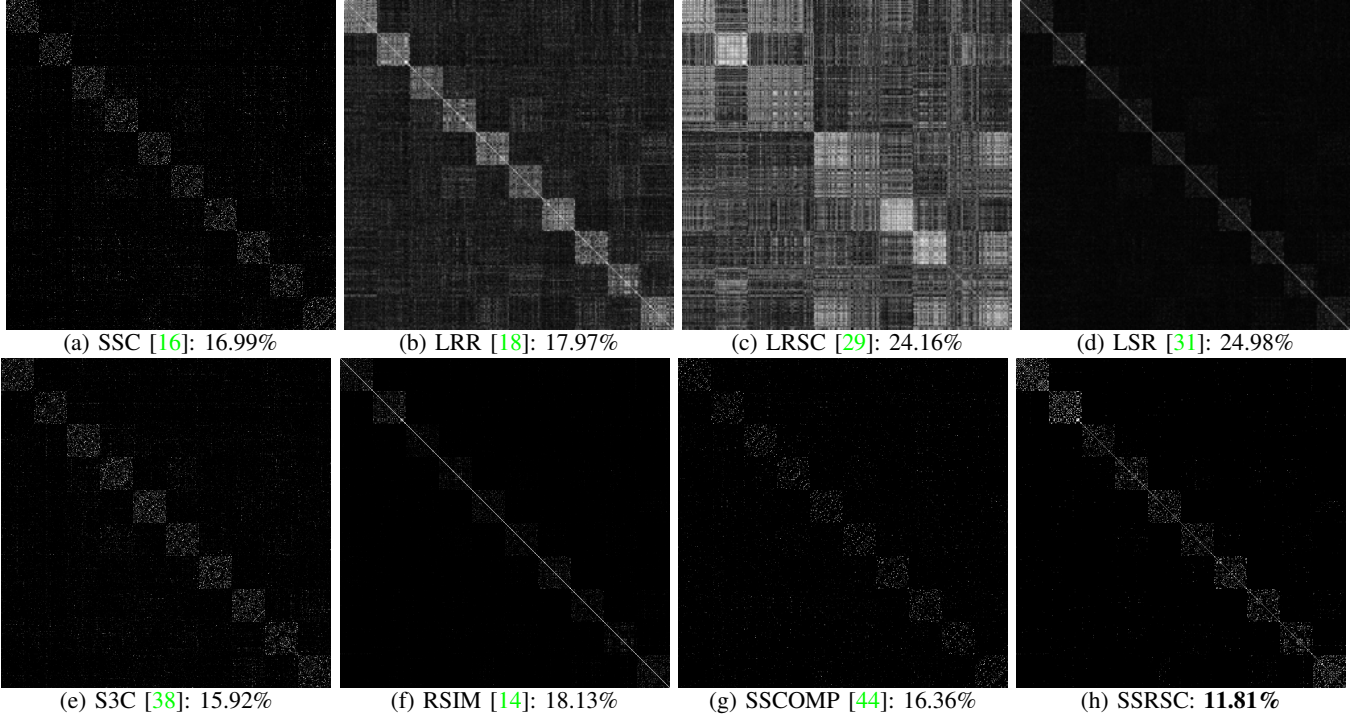


Fig. 6: Affinity matrices and corresponding average clustering errors by different methods on the handwritten digit images from MNIST [50]. 50 images for each digit of $\{0, 1, \dots, 9\}$ are used to compute the affinity matrix by different methods. All images are normalized to $[0, 1]$ by dividing by the maximal entries in the corresponding affinity matrices.

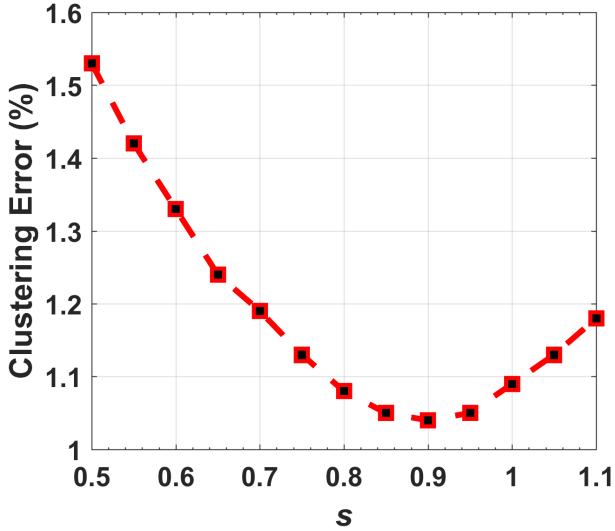


Fig. 7: Average clustering errors (%) of the proposed SSRSC algorithm with different scalar s , on the Hopkins-155 dataset [54].

fast speed of SSRSC owes to both 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 with the Deep Subspace Clustering (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 take the Extended

Yale B dataset [55] for an example. The average clustering errors with respect to the value of s are plotted in Figure 8. As can be seen, SSRSC achieves an average clustering error of 3.26% when $s = 0.5$, and achieves 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 Table II and Table 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 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 SSRSCOMP, and 5.96% for EnSC, respectively. 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 SSRSCOMP, which is the most efficient algorithm among all competing methods. In Table III, we can observe that, the proposed SSRSC method achieves lower clustering error than 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

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

TABLE I: Average clustering errors (%) and speed (in seconds) of different algorithms on the Hopkins-155 dataset [54] with the 12-dimensional data points obtained using PCA. SSRSC can achieve clustering error of 1.04% when $s = 0.9$.

# 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 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.

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 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 clustering error of 21.25% when $s = 0.4$.

# 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 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-dimensional subspace using PCA. The experiments are repeated for 20 times and the average results are reported.

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 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-dimensional subspace using PCA. The experiments are repeated for 10 times and the average results are reported. SSRSC can achieve clustering error of 22.76% when $s = 0.35$.

SSCOMP [44]. On the MNIST [50], the testing data consists 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 consists of $N_i = 500$ images for each of the 26 digits or letters. The feature vectors are extracted from the scattering convolution network [87]. The features extracted from the digit/letter images are originally 3,472-dimensional, and are projected onto a 500-dimensional 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 w.r.t. s is plotted in Figure 9. Average clustering errors are computed over 20 trials sampled from 6,000 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 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 EMNIST dataset, SSRSC achieves clustering error of 23.45%, when the number of images for

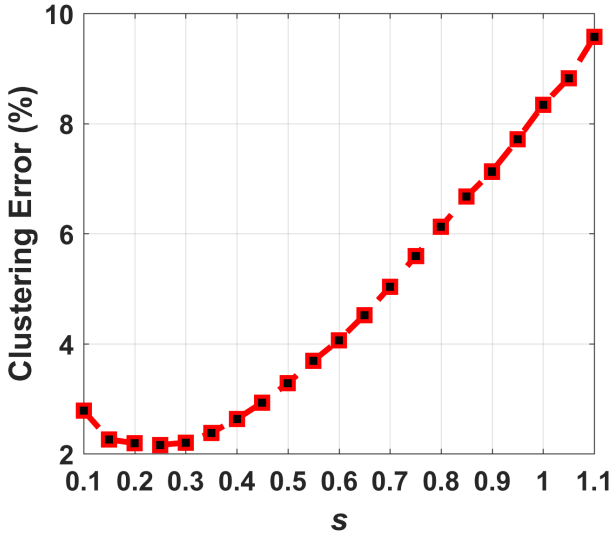


Fig. 8: Average clustering errors (%) of the proposed SSRSC algorithm with different scalar s , on the Extended Yale B dataset [55].

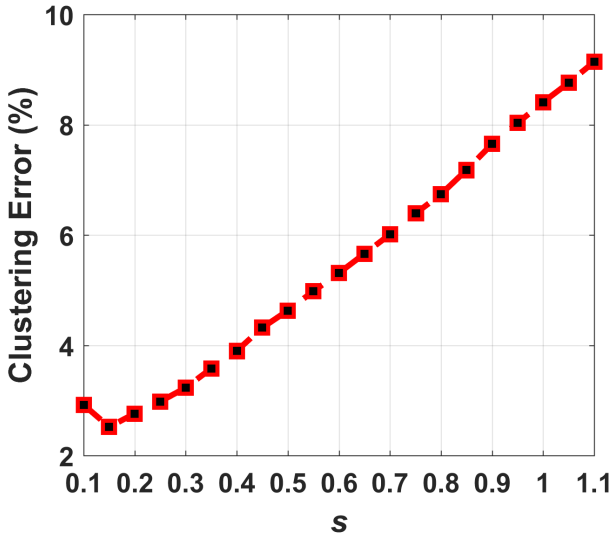


Fig. 9: Average clustering errors (%) of the proposed SSRSC with different s on MNIST [50], with 600 images for each digit.

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 scaled simplex representation over previous sparse or low-rank representation, for hand-written digit/letter clustering.

D. Ablation Study: Effectiveness of Simplex Representation

We perform comprehensive ablation studies on the Hopkins-155 [54], Extended Yale B [55], and MNIST [50] datasets to validate the effectiveness of the proposed SSRSC model. SSRSC includes two constraints, i.e., $c \geq 0$ and $\mathbf{1}^\top c = s$. To analyze the effectiveness of each constraint, we compare the proposed SSRSC model with several baseline methods.

The first baseline is the trivial Least Square Regression (LSR) model as follows:

$$\text{LSR: } \min_C \|X - XC\|_F^2 + \lambda \|C\|_F^2. \quad (20)$$

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 Least Square Regression (NLSR) model:

$$\text{NLSR: } \min_C \|X - XC\|_F^2 + \lambda \|C\|_F^2 \quad \text{s.t.} \quad C \geq 0. \quad (21)$$

The NLSR model can be formulated by either removing the scaled affinity constraint $\mathbf{1}^\top c = s$ from SSRSC or adding a non-negative constraint to the LSR model (20). For structural clearance of this manuscript, we put the solution of the NLSR model (21) in the *Supplementary File*. We can also remove the non-negative constraint $C \geq 0$ in the scaled simplex set $\{C \in \mathbb{R}^{N \times N} | C \geq 0, \mathbf{1}^\top C = s\mathbf{1}^\top\}$, and obtain a Scaled-affine Least Square Regression (SLSR) model:

$$\text{SLSR: } \min_C \|X - XC\|_F^2 + \lambda \|C\|_F^2 \quad \text{s.t.} \quad \mathbf{1}^\top C = s\mathbf{1}^\top. \quad (22)$$

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:

$$\min_C \text{Data Term} + \text{Regularization Term} \quad \text{s.t.} \quad \text{Constraints}. \quad (23)$$

For the NLSR, SLSR, and SSRSC methods, the data and regularization terms are the same as that of the basic LSR model (9), i.e., $\|X - XC\|_F^2$ and $\lambda \|C\|_F^2$, 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 Hopkins-155 [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 Non-negative Constraint. The effectiveness of non-negative constraint $C \geq 0$ 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 NLSR over LSR can directly reflect the effectiveness of the non-negative constraint. Second, the effectiveness of 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.

The results listed in Table VII, show that, on the Hopkins-155 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

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

TABLE VI: Summary of the proposed SSRSC and three baseline methods LSR, NLSR, and SLR.

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

TABLE VII: Average clustering errors (%) of LSR, NLSR, SLR ($s=0.9$), and SSRSC ($s=0.9$) on Hopkins-155 [54], with the 12-dimensional 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

TABLE VIII: Average clustering errors (%) of LSR, NLSR, SLR ($s=0.22$), and SSRSC ($s=0.25$) on Extended Yale B [55], with the $6n$ -dimensional (n is the number of subjects) data points obtained by PCA.

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 Scaled-Affine Constraint. The effectiveness of scaled-affine constraint can be validated from two aspects. First, it can be validated by comparing the baseline methods LSR (20) and SLR (22), which are summarized in Table VI. Since the only difference between them is that SLR contains an additional scaled affine constraint over LSR, the performance gain of SLR 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 Hopkins-155 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 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, i.e., 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

# 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

TABLE IX: Average clustering errors (%) of LSR, NLSR, SLR ($s = 0.24$), and SSRSC ($s = 0.15$) on the MNIST dataset [50]. The features of data point are extracted from a scattering network [87] and projected onto a 500-dimensional subspace obtained using PCA. The experiments are independently repeated 20 times.

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 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 Hopkins-155 dataset and in Table VIII for the Extended Yale B dataset.

Influence of diagonal constraint. One key difference of our SSRSC model to previous models is that, in SSRSC, we get rid of diagonal constraint $\text{diag}(C) = 0$ considering that the practical data is often noisy. To achieve this, we compare with method on the Hopkins155 dataset, together with the SSRSC model with additional diagonal constraint of $\text{diag}(C) = 0$ (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 with a state-of-the-art clustering method, i.e., RGC [88], designed specifically for clustering noisy data. RGC achieves slightly better performance than SSRSC-diag, but is still inferior to our SSRSC without diagonal constraint. This demonstrates the necessity of getting rid of the diagonal constraint in our SSRSC model to deal with noisy data.

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

TABLE X: Average clustering errors (%) of SSRSC ($s = 0.9$), SSRSC-diag ($s = 0.8$) and RGC on Hopkins-155 dataset [54], with the 12-dimensional data points obtained using PCA.

V. CONCLUSION

In this paper, we proposed a scaled simplex representation (SSR) based model for spectral clustering based subspace clustering (SC). Specifically, we introduced the non-negative and

scaled affine constraints into a simple least square regression 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 subspace clustering. Extensive experiments on three benchmark clustering datasets demonstrated that the proposed SSRSC algorithm is very efficient and achieves better clustering performance than state-of-the-art subspace clustering algorithms. The significant improvements of SSRSC over the baseline models demonstrated the effectiveness of the proposed simplex representation.

Our work can be extended in at least three directions. First, it is promising to extend the proposed SSRSC model to nonlinear version by using kernel approaches [60], [69], [70], [89]. Second, it is worthy to accelerate the proposed algorithm for scalable subspace clustering [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 Processing Magazine*, 28(2):52–68, 2011. 1
- [2] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, pages 11–18, 2003. 1, 2
- [3] M. C. Tsakiris and R. Vidal. Algebraic clustering of affine subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):482–489, 2018. 1, 2
- [4] J. Costeira and T. Kanade. A multibody factorization method for independent moving objects. *International Journal of Computer Vision*, 29(3), 1998. 1, 2
- [5] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005. 1, 2
- [6] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999. 1, 2
- [7] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *CVPR*, 2004. 1, 2
- [8] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2
- [9] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1832–1845, 2009. 1, 2
- [10] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. *ECCV*, 2006. 1, 2
- [11] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *CVPR*, 2010. 1, 2
- [12] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. *CVPR*, 2007. 1, 2
- [13] G. Chen and G. Lerman. Spectral curvature clustering. *International Journal of Computer Vision*, 81(3):317–330, 2009. 1, 2, 3, 6
- [14] P. Ji, M. Salzmann, and H. Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. In *ICCV*, pages 4687–4695, 2015. 1, 2, 6, 7, 8, 9
- [15] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797. IEEE, 2009. 1, 2, 3, 5, 6, 7
- [16] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 1, 2, 3, 5, 6, 7, 8, 9
- [17] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010. 1, 2, 3, 6, 7
- [18] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013. 1, 2, 3, 5, 6, 7, 8, 9
- [19] J. Chen and J. Yang. Robust subspace segmentation via low-rank representation. *IEEE Transactions on Cybernetics*, 44(8):1432–1445, 2014. 1
- [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 Transactions on Cybernetics*, 46(8):1828–1838, 2016. 1
- [21] J. Wang, X. Wang, F. Tian, C. H. Liu, and H. Yu. Constrained low-rank representation for robust subspace clustering. *IEEE Transactions on Cybernetics*, 47(12):4534–4546, 2017. 1
- [22] X. Peng, Z. Yu, Z. Yi, and H. Tang. Constructing the l2-graph for robust subspace learning and subspace clustering. *IEEE Transactions on Cybernetics*, 47(4):1053–1066, 2017. 1
- [23] Z. Wen, B. Hou, Q. Wu, and L. Jiao. Discriminative transformation learning for fuzzy sparse subspace clustering. *IEEE Transactions on Cybernetics*, 48(8):2218–2231, 2018. 1
- [24] J. Wen, Y. Xu, and H. Liu. Incomplete multiview spectral clustering with adaptive graph learning. *IEEE Transactions on Cybernetics*, pages 1–12, 2019. 1
- [25] J. Lee, H. Lee, M. Lee, and N. Kwak. Nonparametric estimation of probabilistic membership for subspace clustering. *IEEE Transactions on Cybernetics*, pages 1–14, 2018. 1
- [26] M. Brbi and I. Kopriva. ℓ_0 -motivated low-rank sparse subspace clustering. *IEEE Transactions on Cybernetics*, pages 1–15, 2018. 1
- [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 Transactions on Cybernetics*, 49(5):1722–1734, 2019. 1
- [28] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo. Dual graph regularized latent low-rank representation for subspace clustering. *IEEE Transactions on Image Processing*, 24(12):4918–4933, 2015. 1, 2, 3
- [29] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR*, 2011. 1, 2, 3, 6, 7, 8, 9
- [30] V. Karavasili, K. Blekas, and C. Nikou. A novel framework for motion segmentation and tracking by clustering incomplete trajectories. *Computer Vision and Image Understanding*, 116(11):1135–1148, 2012. 1, 2
- [31] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360. Springer, 2012. 1, 2, 3, 5, 6, 7, 8, 9, 10
- [32] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, pages 3834–3841, 2014. 1, 2, 3, 6, 7, 9
- [33] X. Peng, L. Zhang, and Z. Yi. Scalable sparse subspace clustering. In *CVPR*, June 2013. 1, 2, 3, 5, 6
- [34] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, pages 3818–3825, 2014. 1, 2, 3, 6
- [35] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin. Ordered subspace clustering with block-diagonal priors. *IEEE Transactions on Cybernetics*, 46(12):3209–3219, 2016. 1
- [36] B. Li, Y. Zhang, Z. Lin, and H. Lu. Subspace clustering by mixture of gaussian regression. In *CVPR*, pages 2094–2102, 2015. 1, 2, 3, 6
- [37] C. Li and R. Vidal. Structured sparse subspace clustering: A unified optimization framework. In *CVPR*, pages 277–286, 2015. 1, 2, 3, 5, 6, 7
- [38] C. Li, C. You, and R. Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*, 26(6):2988–3001, June 2017. 1, 2, 3, 6, 7, 8, 9
- [39] J. Xu, K. Xu, K. Chen, and J. Ruan. Reweighted sparse subspace clustering. *Computer Vision and Image Understanding*, 138(0):25–37, 2015. 1, 2, 3, 5, 6
- [40] C. Peng, Z. Kang, and Q. Cheng. Subspace clustering via variance regularized ridge regression. In *CVPR*, 2017. 1, 2, 3, 6
- [41] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In *NIPS*, pages 24–33, 2017. 1, 2, 3, 7, 9
- [42] M. Yin, S. Xie, Z. Wu, Y. Zhang, and J. Gao. Subspace clustering via learning an adaptive low-rank graph. *IEEE Transactions on Image Processing*, 27(8):3716–3728, Aug 2018. 1, 2
- [43] J. Yang, J. Liang, K. Wang, Y. L. Yang, and M. M. Cheng. Automatic model selection in subspace clustering via triplet relationships. In *AAAI*, 2018. 1, 2
- [44] C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *CVPR*, pages 3918–3927, 2016. 1, 2, 3, 6, 7, 8, 9
- [45] C. You, C. Li, D. P. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *CVPR*, pages 3928–3937, 2016. 1, 3, 7, 9
- [46] C. You, D. P. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. In *CVPR*, pages 1–10, 2017. 1, 2

- [47] C. You, C. Li, D. P. Robinson, and R. Vidal. Scalable exemplar-based subspace clustering on class-imbalanced data. In *ECCV*, September 2018. 1, 2, 7, 9, 12
- [48] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 1
- [49] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 1
- [50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 6, 7, 8, 9, 10, 11
- [51] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. 2
- [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 *CVPR*, pages 2328–2335. IEEE, 2012. 2
- [53] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, January 2011. 2, 4, 6
- [54] R. Tron and R. Vidal. A benchmark for the comparison of 3d motion segmentation algorithms. In *CVPR*, 2007. 2, 5, 6, 7, 8, 9, 10, 11
- [55] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001. 2, 6, 7, 8, 9, 10, 11
- [56] A. Ellis and J. Ferryman. Biologically-inspired robust motion segmentation using mutual information. *Computer Vision and Image Understanding*, 122:47–64, 2014. 2
- [57] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transaction on Information Theory*, 52(2):489–509, 2006. 3
- [58] D. Donoho. Compressed sensing. *IEEE Transaction on Information Theory*, 52(4):1289–1306, 2006. 3
- [59] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 3, 6
- [60] C. Peng, Z. Kang, and Q. Cheng. Integrating feature and graph learning with low-rank representation. *Neurocomputing*, 249:106–116, 2017. 3, 12
- [61] X. Peng, Z. Yi, and H. Tang. Robust subspace clustering via thresholding ridge regression. In *AAAI*, 2015. 3
- [62] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. *IEEE International Conference on Computer Vision (ICCV)*, pages 244–252, 2015. 3
- [63] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017. 3
- [64] J. Xu, L. Zhang, and D. Zhang. External prior guided internal prior learning for real-world noisy image denoising. *IEEE Transactions on Image Processing*, 27(6):2996–3010, 2018. 3
- [65] Zhetong Liang, Jun Xu, David Zhang, Zisheng Cao, and Lei Zhang. A hybrid 11-10 layer decomposition model for tone mapping. In *CVPR*, June 2018. 3
- [66] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018. 3
- [67] C. Peng, Z. Kang, H. Li, and Q. Cheng. Subspace clustering using log-determinant rank approximation. In *SIGKDD*, pages 925–934. ACM, 2015. 3
- [68] L. Zhuang, S. Gao, J. Tang, J. Wang, Z. Lin, Y. Ma, and N. Yu. Constructing a nonnegative low-rank and sparse graph with data-adaptive features. *IEEE Transactions on Image Processing*, 24(11):3717–3728, 2015. 3
- [69] Z. Kang, C. Peng, and Q. Cheng. Kernel-driven similarity learning. *Neurocomputing*, 267:210–219, 2017. 3, 12
- [70] Z. Kang, L. Wen, W. Chen, and Z. Xu. Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems*, 163:510–517, 2019. 3, 12
- [71] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu. Clustering with similarity preserving. *Neurocomputing*, 2019. 3
- [72] J. Xu, W. An, L. Zhang, and D. Zhang. Sparse, collaborative, or nonnegative representation: Which helps pattern classification? *Pattern Recognition*, 88:679–688, 2019. 3
- [73] Q. Li, W. Liu, and L. Li. Affinity learning via a diffusion process for subspace clustering. *Pattern Recognition*, 84:39–50, 2018. 3
- [74] J. Huang, F. Nie, and H. Huang. A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI*, 2015. 3
- [75] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1):1–23, 1943. 4
- [76] J. Eckstein and D. P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992. 4
- [77] K. Riedel. A sherman-morrison-woodbury identity for rank augmenting matrices with application to centering. *SIAM Journal on Matrix Analysis and Applications*, 13(2):659–662, 1992. 4
- [78] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. 4
- [79] E. Elhamifar, G. Sapiro, and S. Sastry. Dissimilarity-based sparse subset selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2182–2197, 2016. 4
- [80] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of \mathbb{R}^n . *Journal of Optimization Theory and Applications*, 50(1):195–200, July 1986. 4
- [81] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *ICML*, pages 272–279. ACM, 2008. 4
- [82] L. Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, 158(1):575–585, Jul 2016. 4
- [83] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001. 6
- [84] F. S. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994. 6, 7, 8, 9
- [85] G. Cohen, S. Afshar, J. Tapson, and A. Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017. 6, 7, 8, 9
- [86] I. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*, pages 115–128. Springer, 1986. 6
- [87] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013. 7, 9, 11
- [88] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, 2019. 11
- [89] C. Peng and Q. Cheng. Discriminative regression machine: A classifier for high-dimensional data or imbalanced data, 2019. 12
- [90] C. Peng, C. Chen, Z. Kang, J. Li, and Q. Cheng. Res-pca: A scalable approach to recovering low-rank matrices. In *CVPR*, June 2019. 12