




Robust Subspace Clustering via Latent Smooth Representation Clustering

Xiaobo Xiao¹ · Lai Wei¹ 

Published online: 27 July 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Subspace clustering aims to group high-dimensional data samples into several subspaces which they were generated. Among the existing subspace clustering methods, spectral clustering-based algorithms have attracted considerable attentions because of their predominant performances shown in many subspace clustering applications. In this paper, we proposed to apply smooth representation clustering (SMR) to the reconstruction coefficient vectors which were obtained by sparse subspace clustering (SSC). Because the reconstruction coefficient vectors could be regarded as a kind of good representations of original data samples, the proposed method could be considered as a SMR performed in a latent subspace found by SSC and hoped to achieve better performances. For solving the proposed latent smooth representation algorithm (LSMR), we presented an optimization method and also discussed the relationships between LSMR with some related algorithms. Finally, experiments conducted on several famous databases demonstrate that the proposed algorithm dominates the related algorithms.

Keywords Subspace clustering · Smooth representation · Sparse representation · Latent subspace

1 Introduction

The high-dimensional data samples in computer vision and pattern recognition applications are usually viewed as lying in/near several low-dimensional subspaces [1, 2]. Subspace clustering algorithms attempt to partition the high-dimensional data samples into the corresponding subspaces where they are drawn from [3–8]. Among the existing subspace clustering methods, methods based on spectral clustering [3, 5, 9, 10] have attracted lots of interests due to their prominent performances in many machine learning and pattern recognition tasks.

✉ Lai Wei
weilai@shmtu.edu.cn

Xiaobo Xiao
xiaobo_xiao@126.com

¹ Department of Computer Science, Shanghai Maritime University, Haigang Avenue 1550, Shanghai 201306, China

Typically, spectral clustering-based algorithms divide the subspace clustering processes into three steps: (1) using the given data samples to compute a reconstructive coefficient matrix \mathbf{Z} , (2) applying the obtained coefficient matrix \mathbf{Z} to construct an affinity graph \mathbf{W} , (3) adopting a kind of spectral clustering (e.g. Normalize cuts (N-cut) [11] to produce the clustering results. It is known that the performance of a spectral clustering-based method greatly depends on whether the computed reconstructive coefficient matrix \mathbf{Z} could reveal the intrinsic structure of a data set.

1.1 Related Work

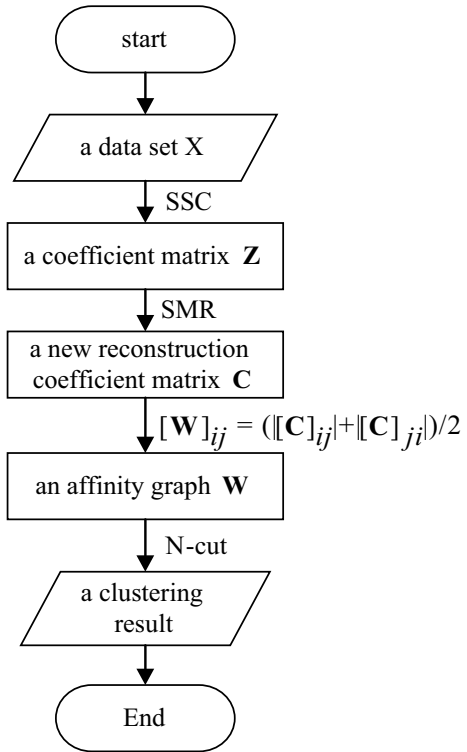
As a result, the main efforts of spectral clustering-based methods concentrate on the construction of coefficient matrices [5, 6, 12, 13]. The two most representative algorithms of spectral clustering-based algorithms are sparse subspace clustering (SSC) [3, 4] and low-rank representation (LRR) [5, 6]. A large amount of subsequent researches have been put forward [14–23] as a consequence of the excellent performances showed by SSC and LRR.

For instance, Li et al. presented a structured SSC method (SSSC) [14, 15] in which they designed an adaptive weighted sparse constraint for the reconstructive coefficient matrix obtained by SSC. Chen et al. introduced a new within-class grouping constraint of the reconstructive coefficient matrix into SSSC [16]. Zhuang et al. presented a non-negative low rank and sparse representation (NNLRSR) method [17] in which they added sparse and non-negative constraints into LRR. Tang et al. analyzed NNLRSR method and devised a structure-constrained LRR algorithm (SCLRR) which they declared to be an extension of NNLRSR [18]. Wei et al. proposed a spectral clustering steered low-rank representation method (SCSLRR) [19] by summarizing the subspace clustering procedures of LRR-based algorithms. They showed that SCSLRR could be considered as a generalization of SCLRR. Lu et al. suggested a graph-regularized LRR model (GLRR) [20] which added Laplacian regularization to LRR in that they argued that adjacent samples should have similar coefficient vectors. Yin et al. designed a non-negative sparse Laplacian regularized LRR model (NSLLRR) [21] which combined GLRR and NNLRSR. What's more, NSLLRR could be treated as an extension of GLRR.

Additionally, we find that some algorithms obtained compelling results by minimizing different norms of the coefficient matrix as well. For example, Lu et al. intended to minimize the Frobenius norms of a coefficient matrix and proposed a least square regression (LSR) [22]. Hu et al. presented a kind of smooth representation clustering (SMR) algorithm [23] which expected coefficient matrices to retain the locality structures of an original data set. We could see the above mentioned algorithms followed the same methodology as we mentioned in the second paragraph for handling subspace clustering problems.

What's more, Yu et al. proposed a novel ranking model based on the learning-to-rank framework where visual features and click features were simultaneously utilized to obtain the ranking model [24]. Hong et al. presented a novel approach which could recover 3-D human poses from silhouettes [25]. This approach improved traditional methods by adopting multi-view locality-sensitive sparse coding in the retrieving process. Hong et al. used multimodal data and devised a novel face-pose estimation framework named multi-task manifold deep learning (M2DL) [26]. Zhang et al. presented an unsupervised deep-learning framework named local deep-feature alignment (LDFA) for dimension reduction which could learn both local and global characteristics of the data sample set [27]. Yu et al. devised a Hierarchical Deep Word Embedding (HDWE) model by integrating sparse constraints and an improved RELU operator to address click feature prediction from visual

Fig. 1 The clustering procedure of LSMR



features [28]. Yu et al. proposed an end-to-end place recognition model based on a novel deep neural network [29]. Yu et al. designed a one-stage framework, SPRNet, which performed efficient instance segmentation by introducing a single pixel reconstruction (SPR) branch to off-the-shelf one-stage detectors [30].

1.2 Contributions

In this paper, we proposed a new subspace clustering algorithm, termed latent smooth representation clustering (LSMR). In LSMR, for a data set \mathbf{X} , we firstly obtained the coefficient matrix \mathbf{Z} by using SSC, then \mathbf{Z} could be viewed as the representations of the original data set \mathbf{X} in a latent subspace. Secondly, in this latent space, SMR was performed on \mathbf{Z} to obtain a new reconstruction coefficient matrix \mathbf{C} . Finally, \mathbf{C} would be used to construct the affinity graph \mathbf{W} and the subspace clustering result could be produced. Compared with the existing spectral clustering-based algorithms, LSMR uses a totally different method to obtain the affinity graph and our experiment results show that LSMR outperforms the existing related algorithms. Figure 1 shows the clustering procedure of LSMR.

The rest of this paper is organized as follows: Sect. 2 briefly reviews SSC algorithms. Section 3 introduces the idea of LSMR method. And the optimization method for solving LSMR is presented in this section as well. We further discuss the complexity of LSMR and its relationships with some related algorithms in Sect. 4. The extensive experiments performed to illuminate the effectiveness of LSMR are showed in Sect. 5. Section 6 gives the conclusions.

2 Sparse Subspace Clustering (SSC)

In this section, we briefly recap the sparse subspace clustering (SSC) algorithm.

For a data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in R^{d \times n}$ with n data samples, sparse subspace clustering (SSC) expects to find a reconstruction coefficient matrix $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n] \in R^{n \times n}$. \mathbf{Z} satisfies $\mathbf{X} = \mathbf{XZ} + \mathbf{E}$ where $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] \in R^{d \times n}$ indicates the reconstruction residual matrix. SSC hopes to minimize the l_1 -norms of \mathbf{Z} and \mathbf{E} . Hence the objective function of SSC could be precisely expressed as the following Problem (1):

$$\begin{aligned} & \min_{\mathbf{Z}} \|\mathbf{Z}\|_1 + \lambda \|\mathbf{E}\|_1 \\ & s.t. \mathbf{X} = \mathbf{XZ} + \mathbf{E} \\ & [\mathbf{Z}]_{ii} = 0, i = 1, 2, \dots, n, \end{aligned} \tag{1}$$

where $\lambda > 0$ is a parameter which is used to balance the effects of the two terms and $[\mathbf{Z}]_{ij}$ represents the (i, j) -th element of \mathbf{Z} . SSC uses the alternating direction method (ADM) [31] to address Problem (1). Then the solution \mathbf{Z} is used to construct an affinity graph \mathbf{W} which satisfies $[\mathbf{W}]_{ij} = (|[\mathbf{Z}]_{ij}| + |[\mathbf{Z}]_{ji}|)/2$. Finally, the final clustering result could be obtained by using N-cut.

3 Latent Smooth Representation Clustering

3.1 Motivation

In fact, SSC considers each sample $\mathbf{x}_i \in \mathbf{X}$ can be linearly reconstructed by using the rest samples as few as possible in \mathbf{X} [32]. Precisely, $\mathbf{x}_i = \mathbf{Xz}_i + \mathbf{e}_i$, where $\mathbf{z}_i \in \mathbf{Z}$ is a sparse reconstruction coefficient vector corresponding to \mathbf{x}_i and \mathbf{e}_i is the corresponding reconstruction error. According to the descriptions, \mathbf{z}_i could be regarded as a representation of \mathbf{x}_i . In real applications, the dimensionality of data is usually very high, i.e., $d \gg n$. Then the found representation $\mathbf{Z} \in R^{n \times n}$ could be actually regarded as the low-dimensional embedding of the original data matrix \mathbf{X} in a latent subspace found by SSC.

In the classical SSC, \mathbf{Z} will be used to construct the affinity graph directly. However, from the viewpoint described above, we could perform subspace clustering in the latent subspace obtained by SSC. Here many existing subspace clustering algorithms could be applied, we choose smooth representation clustering (SMR) method to get the final clustering results for its efficiency and effectiveness.

The objective function of classical SMR in original observed space could be expressed as follows:

$$\min_{\mathbf{C}} \beta \|\mathbf{X} - \mathbf{XC}\|_F^2 + tr(\mathbf{CL}_X\mathbf{C}^t), \tag{2}$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n] \in R^{n \times n}$ is the reconstruction coefficient matrix corresponding to data matrix \mathbf{X} , \mathbf{C}^t denotes the transpose of \mathbf{C} , $\|\cdot\|_F$ denotes the Frobenius norm and $\beta > 0$ is a parameter. $tr(\mathbf{CL}_X\mathbf{C}^t) = \sum_{i=1}^n \sum_{j=1}^n [\mathbf{W}_X]_{ij} \|\mathbf{c}_i - \mathbf{c}_j\|_2^2$ is a Laplacian regularization term which is adopted to enforce the grouping effect¹ of \mathbf{C} . \mathbf{L}_X is the Laplacian matrix defined

¹ Namely, if $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \rightarrow 0$, we have $\|\mathbf{c}_i - \mathbf{c}_j\|_2^2 \rightarrow 0$.

by $\mathbf{L}_X = \mathbf{D}_x - \mathbf{W}_x$, in which \mathbf{D}_x is the diagonal matrix with $[\mathbf{D}_x]_{ii} = \sum_{j=1}^n [\mathbf{W}_x]_{ij}$. \mathbf{W}_x is the similarity matrix measuring the spatial closeness of samples and constructed by using the k nearest neighbors (knn) [33] graph. Namely, if x_j is one of the k nearest neighbors of x_i , $[\mathbf{W}_x]_{ij} = 1$. $[\mathbf{W}_x]_{ij} = 0$, otherwise.

Based on our interpretation previously, SMR could be easily transferred to the latent subspace found by SSC for a given data set \mathbf{X} . And in the latent subspace, \mathbf{Z} is the representation of \mathbf{X} . Consequently, the objective function of SMR could be transferred to be:

$$\min_{\mathbf{C}} \beta \|\mathbf{Z} - \mathbf{ZC}\|_F^2 + tr(\mathbf{CL}_Z\mathbf{C}^t), \tag{3}$$

where $tr(\mathbf{CL}_Z\mathbf{C}^t) = \sum_{i=1}^n \sum_{j=1}^n [\mathbf{W}_Z]_{ij} \|\mathbf{c}_i - \mathbf{c}_j\|_2^2$ and $\mathbf{L}_Z = \mathbf{D}_Z + \mathbf{W}_Z$, in which \mathbf{D}_Z is a diagonal matrix with $[\mathbf{D}_Z]_{ii} = \sum_{j=1}^n [\mathbf{W}_Z]_{ij}$. Because \mathbf{Z} is a reconstruction coefficient matrix with $[\mathbf{Z}]_{ij}$ measuring the similarity between original data points x_i and x_j , we could define \mathbf{W}_Z as $[\mathbf{W}_Z]_{ij} = (|[\mathbf{Z}]_{ij}| + |[\mathbf{Z}]_{ji}|)/2$.

Here, researchers may argue that $[\mathbf{W}_Z]_{ij}$ does not indicate the similarity between z_i and z_j . However, as described above, \mathbf{Z} is the representation of \mathbf{X} , so the similarity between original data points x_i and x_j should be close to the similarity between z_i and z_j , thus the definition of \mathbf{W}_Z is reasonable. Moreover, in SMR, the neighborhood scale k is usually difficult to choose for different data sets and the similarities between pairwise data samples are manually set. This would make the performance of SMR unstable. In our method, we could bypass these disadvantages.

Finally, we combine Problem (1) and Problem (3) together and let $\mathbf{E}_1 = \mathbf{X} - \mathbf{XZ}$, $\mathbf{E}_2 = \mathbf{Z} - \mathbf{ZC}$, then the following problem could be have, i.e.:

$$\begin{aligned} & \min_{\mathbf{Z}, \mathbf{C}, \mathbf{E}_1, \mathbf{E}_2} \|\mathbf{Z}\|_1 + \alpha \|\mathbf{E}_1\|_1 + tr(\mathbf{CL}_Z\mathbf{C}^t) + \beta \|\mathbf{E}_2\|_F^2, \\ & s.t. \mathbf{E}_1 = \mathbf{X} - \mathbf{XZ}, \\ & \mathbf{E}_2 = \mathbf{Z} - \mathbf{ZC}, \\ & [\mathbf{Z}]_{ii} = 0, i = 1, 2, \dots, n, \end{aligned} \tag{4}$$

where α, β are two positive parameters. In real applications, $\tilde{\mathbf{L}}_Z = \mathbf{L}_Z + \epsilon \mathbf{I}_n$ is often used instead of \mathbf{L}_Z , where \mathbf{I}_n is an $n \times n$ identity matrix and ϵ is a small positive parameter.² Then $\tilde{\mathbf{L}}_Z$ is enforced to be strictly positive defined. We call Problem (4) latent smooth representation clustering (LSMR). Problem (4) could be solved by using the alternating direction method (ADM) [31].

3.2 Optimization

For addressing Problem (4), first of all, we covert it into the following equivalent formulation:

² In our experiments, $\epsilon = 0.01$.

$$\begin{aligned}
 & \min_{\mathbf{Z}, \mathbf{C}, \mathbf{M}, \mathbf{J}, \mathbf{E}_1, \mathbf{E}_2} \|\mathbf{M}\|_1 + \alpha \|\mathbf{E}_1\|_1 + \text{tr}(\mathbf{Q}\tilde{\mathbf{L}}_{\mathbf{Z}}\mathbf{Q}^t) + \beta \|\mathbf{E}_2\|_F^2 \\
 & \text{s.t. } \mathbf{E}_1 = \mathbf{X} - \mathbf{XZ}, \\
 & \quad \mathbf{Z} = \mathbf{M} \\
 & \quad [\mathbf{M}]_{ii} = 0, i = 1, 2, \dots, n, \\
 & \quad \mathbf{E}_2 = \mathbf{Z} - \mathbf{ZC}, \\
 & \quad \mathbf{C} = \mathbf{Q}.
 \end{aligned} \tag{5}$$

Then the augmented Lagrangian function of Problem (5) can be defined as follows:

$$\begin{aligned}
 \mathcal{L} = & \|\mathbf{M}\|_1 + \alpha \|\mathbf{E}_1\|_1 + \text{tr}(\mathbf{Q}\tilde{\mathbf{L}}_{\mathbf{Z}}\mathbf{Q}^t) + \beta \|\mathbf{E}_2\|_F^2 + \langle \mathbf{Y}_1, \mathbf{X} - \mathbf{XZ} - \mathbf{E}_1 \rangle \\
 & + \langle \mathbf{Y}_2, \mathbf{Z} - \mathbf{M} \rangle + \langle \mathbf{Y}_3, \mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 \rangle + \langle \mathbf{Y}_4, \mathbf{C} - \mathbf{Q} \rangle \\
 & + \mu/2(\|\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1\|_F^2 + \|\mathbf{Z} - \mathbf{M}\|_F^2 \\
 & + \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2\|_F^2 + \|\mathbf{C} - \mathbf{Q}\|_F^2)
 \end{aligned} \tag{6}$$

where $\mu > 0$ is a parameter, $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$ and \mathbf{Y}_4 are four Lagrange multipliers. After that, the variables $\mathbf{Z}, \mathbf{C}, \mathbf{M}, \mathbf{Q}, \mathbf{E}_1$ and \mathbf{E}_2 could be optimized alternately with minimizing \mathcal{L} . The detailed updating procedure for each variable is showed in the following sections:

1. **Update M with fixed other variables.** By picking the relevant terms of \mathbf{M} in Eq. (6), we have:

$$\begin{aligned}
 & \min_{\mathbf{M}} \|\mathbf{M}\|_1 + \langle \mathbf{Y}_2, \mathbf{Z} - \mathbf{M} \rangle + \mu/2\|\mathbf{Z} - \mathbf{M}\|_F^2 \\
 & = \min_{\mathbf{M}} \|\mathbf{M}\|_1 + \mu/2\|\mathbf{Z} - \mathbf{M} + \frac{\mathbf{Y}_2}{\mu}\|_F^2,
 \end{aligned} \tag{7}$$

then the solution to Eq. (7) could be obtained as

$$[\mathbf{M}^{opt}]_{ij} = \begin{cases} \max\left(0, [\mathbf{Z} + \mathbf{Y}_2/\mu]_{ij} - 1/\mu\right) + \min\left(0, [\mathbf{Z} + \mathbf{Y}_2/\mu]_{ij} + 1/\mu\right), & i \neq j; \\ 0 & \end{cases} \tag{8}$$

\mathbf{M}^{opt} denotes the optimal value of \mathbf{M} .

2. **Update Z with fixed other variables.** By abandoning the irrelevant terms of \mathbf{C} in Eq. (6), we have:

$$\begin{aligned}
 & \min_{\mathbf{C}} \langle \mathbf{Y}_1, \mathbf{X} - \mathbf{XZ} - \mathbf{E}_1 \rangle + \langle \mathbf{Y}_2, \mathbf{Z} - \mathbf{M} \rangle + \langle \mathbf{Y}_3, \mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 \rangle \\
 & + \mu/2(\|\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1\|_F^2 + \|\mathbf{Z} - \mathbf{M}\|_F^2 + \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2\|_F^2) \\
 & = \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1 + \mathbf{Y}_1/\mu\|_F^2 + \|\mathbf{Z} - \mathbf{M} + \mathbf{Y}_2/\mu\|_F^2 \\
 & + \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 + \mathbf{Y}_3/\mu\|_F^2.
 \end{aligned} \tag{9}$$

We take the derivation of Eq. (9) w.r.t. \mathbf{Z} and set it to 0, the following equation holds:

$$\begin{aligned}
 & (\mathbf{X}^t \mathbf{X} + \mathbf{I}_n) \mathbf{Z}^{opt} + \mathbf{Z}^{opt} (\mathbf{I}_n - \mathbf{C}) (\mathbf{I}_n - \mathbf{C}^t) - \mathbf{X}^t (\mathbf{X} - \mathbf{E}_1 + \mathbf{Y}_1 / \mu) \\
 & - \mathbf{M} + \mathbf{Y}_2 / \mu - \left(\mathbf{E}_2 - \frac{\mathbf{Y}_3}{\mu} \right) (\mathbf{I}_n - \mathbf{C}^t) = 0.
 \end{aligned} \tag{10}$$

Equation (10) is a Sylvester equation [34] w.r.t. \mathbf{Z}^{opt} , so it can be solved by the Matlab function `lyap()`. It should be pointed out that though $\tilde{\mathbf{L}}_{\mathbf{Z}}$ is designed by \mathbf{Z} , we treat $\tilde{\mathbf{L}}_{\mathbf{Z}}$ as a fixed value when \mathbf{Z} is updating. After \mathbf{Z} is updated, then $\tilde{\mathbf{L}}_{\mathbf{Z}}$ would be computed.

3. **Update Q with fixed other variables.** By collecting the relevant terms of \mathbf{Q} , we have:

$$\begin{aligned}
 & \min_{\mathbf{Q}} tr(\mathbf{Q} \tilde{\mathbf{L}}_{\mathbf{Z}} \mathbf{Q}^t) + \langle \mathbf{Y}_4, \mathbf{C} - \mathbf{Q} \rangle + \mu/2 \|\mathbf{C} - \mathbf{Q}\|_F^2 \\
 & = \min_{\mathbf{Q}} tr(\mathbf{Q} \tilde{\mathbf{L}}_{\mathbf{Z}} \mathbf{Q}^t) + \mu/2 \|\mathbf{C} - \mathbf{Q} + \frac{\mathbf{Y}_4}{\mu}\|_F^2.
 \end{aligned} \tag{11}$$

By taking the derivation of Eq. (11) w.r.t. \mathbf{Q} and setting it to 0, then the following equation holds:

$$\mathbf{Q}^{opt} = (\mu \mathbf{C} + \mathbf{Y}_4) (2 \tilde{\mathbf{L}}_{\mathbf{Z}} + \mu \mathbf{I}_n)^{-1}. \tag{12}$$

4. **Update C with fixed other variables.** By collecting the relevant terms w.r.t \mathbf{C} , minimizing Eq. (6) becomes to the following problem:

$$\begin{aligned}
 & \min_{\mathbf{Z}} \langle \mathbf{Y}_3, \mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 \rangle + \langle \mathbf{Y}_4, \mathbf{C} - \mathbf{Q} \rangle + \mu/2 (\|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2\|_F^2 + \|\mathbf{C} - \mathbf{Q}\|_F^2) \\
 & = \min_{\mathbf{Z}} \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 + \mathbf{Y}_3 / \mu\|_F^2 + \|\mathbf{C} - \mathbf{Q} + \mathbf{Y}_4 / \mu\|_F^2
 \end{aligned} \tag{13}$$

By applying the same methods used for updating \mathbf{Q} , the following equation holds:

$$\mathbf{C}^{opt} = (\mathbf{Z}^t \mathbf{Z} + \mathbf{I}_n)^{-1} [\mathbf{Z}^t (\mathbf{Z} - \mathbf{E}_2 + \mathbf{Y}_3 / \mu) + \mathbf{Q} - \mathbf{Y}_4 / \mu] \tag{14}$$

5. **Update E1 with fixed other variables.** By dropping the irrelevant terms of \mathbf{E}_1 , afterward minimizing Eq. (6) equals addressing the following problem:

$$\begin{aligned}
 & \min_{\mathbf{E}_1} \alpha \|\mathbf{E}_1\|_1 + \langle \mathbf{Y}_1, \mathbf{X} - \mathbf{XZ} - \mathbf{E}_1 \rangle + \mu/2 \|\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1\|_F^2 \\
 & = \min_{\mathbf{E}_1} \alpha \|\mathbf{E}_1\|_1 + \mu/2 \|\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1 + \mathbf{Y}_1 / \mu\|_F^2
 \end{aligned} \tag{15}$$

Similar to computing the optimal value of \mathbf{M} , we could obtain:

$$[\mathbf{E}_1^{opt}]_{ij} = \max\left(0, [\mathbf{X} - \mathbf{XZ} + \mathbf{Y}_1 / \mu]_{ij} - \alpha / \mu\right) + \min\left(0, [\mathbf{X} - \mathbf{XZ} + \mathbf{Y}_1 / \mu]_{ij} + \alpha / \mu\right) \tag{16}$$

6. **Update E2 with fixed other variables.** By gathering the relevant terms of \mathbf{E}_2 in Eq. (6), we have:

$$\begin{aligned}
 & \min_{\mathbf{E}_2} \beta \|\mathbf{E}_2\|_F^2 + \langle \mathbf{Y}_3, \mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 \rangle + \mu/2 \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2\|_F^2 \\
 & = \min_{\mathbf{E}_2} \beta \|\mathbf{E}_2\|_F^2 + \mu/2 \|\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2 + \frac{\mathbf{Y}_3}{\mu}\|_F^2.
 \end{aligned} \tag{17}$$

We also take the derivation of Eq. (17) w.r.t. \mathbf{E}_2 and set it to 0, subsequently the following equation holds:

$$\mathbf{E}_2^{opt} = \frac{\mu}{(2\beta + \mu)\left(\mathbf{Z} - \mathbf{ZC} + \frac{\mathbf{Y}_3}{\mu}\right)}. \tag{18}$$

7. **Update parameters with fixed other variables.** The precise updating schemes for the parameters existed in Eq. (6) are summarized as follows:

$$\begin{aligned} \mathbf{Y}_1^{opt} &= \mathbf{Y}_1 + \mu(\mathbf{X} - \mathbf{XZ} - \mathbf{E}_1), \\ \mathbf{Y}_2^{opt} &= \mathbf{Y}_2 + \mu(\mathbf{Z} - \mathbf{M}), \\ \mathbf{Y}_3^{opt} &= \mathbf{Y}_3 + \mu(\mathbf{Z} - \mathbf{ZC} - \mathbf{E}_2), \\ \mathbf{Y}_4^{opt} &= \mathbf{Y}_4 + \mu(\mathbf{C} - \mathbf{Q}), \\ \mu^{opt} &= \min(\mu_{max}, \rho\mu), \end{aligned} \tag{19}$$

where μ_{max} and ρ are two given positive parameters.

3.3 Algorithm

The algorithmic procedure of LSMR is summarized in Algorithm 1. Once the solutions to LSMR are gotten, LSMR defines an affinity graph \mathbf{W} satisfying $[\mathbf{W}]_{ij} = (|[\mathbf{C}]_{ij}| + |[\mathbf{C}]_{ji}|)/2$. Finally, we could obtain clustering results by N-cut performed on the graph.

Algorithm 1. Latent Smooth Representation Clustering (LSMR)

Input:
 Data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in R^{d \times n}$, parameters α, β , the maximal number of iteration *Maxiter*;

Output:
 The two coefficient matrices $\mathbf{Z}^{opt}, \mathbf{C}^{opt}$, and two noise term $\mathbf{E}_1^{opt}, \mathbf{E}_2^{opt}$.

- 1: Initialize the parameters and variables, i.e., $\mathbf{Y}_1^0 = \mathbf{Y}_2^0 = \mathbf{Y}_3^0 = \mathbf{Y}_4^0 = \mathbf{0}, \mu^0 = 10^{-1}, \mu_{max} = 10^{30}, \rho = 1.1, \varepsilon = 10^{-4}, k = 0, \mathbf{Z}^0 = \mathbf{Q}^0 = \mathbf{C}^0 = \mathbf{0}$.
- 2: **while** $\|\mathbf{X} - \mathbf{XZ}^k - \mathbf{E}_1^k\|_\infty > \varepsilon, \|\mathbf{Z}^k - \mathbf{M}^k\|_\infty > \varepsilon, \|\mathbf{Z}^k - \mathbf{Z}^k\mathbf{C}^k - \mathbf{E}_2^k\|_\infty > \varepsilon, \|\mathbf{C}^k - \mathbf{Q}^k\|_\infty > \varepsilon$, and $k < \text{Maxiter}$ **do**
- 3: Update $[\mathbf{M}]_{ij}^{k+1}$ by using Eq. (8);
- 4: Update \mathbf{Z}^{k+1} by using Matlab function *lyap()* to solve Eq. (10);
- 5: Update \mathbf{Q}^{k+1} by solving Eq. (12);
- 6: Update \mathbf{C}^{k+1} by solving Eq. (14);
- 7: Update $[\mathbf{E}_1]_{ij}^{k+1}$ by using Eq. (16);
- 8: Update $[\mathbf{E}_2]_{ij}^{k+1}$ by solving Eq. (18);
- 9: Update $\mathbf{Y}_1^{k+1}, \mathbf{Y}_2^{k+1}, \mathbf{Y}_3^{k+1}, \mathbf{Y}_4^{k+1}, \mu^{k+1}$ by following the updating schedule in Eq. (19);
- 10: set $k = k + 1$;
- 11: **end while**
- 12: **return** the coefficient matrix $\mathbf{Z}^{opt} = \mathbf{Z}^k, \mathbf{C}^{opt} = \mathbf{C}^k, \mathbf{E}_1^{opt} = \mathbf{E}_1^k, \mathbf{E}_2^{opt} = \mathbf{E}_2^k$.

4 Further Discussions

4.1 Complexity Analysis

We will discuss the computational complexity of Algorithm 1 in this subsection. The running time of Algorithm 1 is mainly composed by the updating of six variables: $\mathbf{M} \in R^{n \times n}, \mathbf{Z} \in R^{n \times n}, \mathbf{Q} \in R^{n \times n}, \mathbf{C} \in R^{n \times n}, \mathbf{E}_1 \in R^{d \times n}, \mathbf{E}_2 \in R^{d \times n}$. Then we will analyze the computational burden of updating $\mathbf{M}, \mathbf{Z}, \mathbf{Q}, \mathbf{C}, \mathbf{E}_1, \mathbf{E}_2$ in each step.

Firstly, updating \mathbf{M} and \mathbf{E}_1 both need to compute each element of two $n \times n$ matrices, therefore their time complexities are $O(n^2)$. Secondly, updating \mathbf{Z} needs to solve a Sylvester equation, which takes $O(n^3)$ time. Thirdly, it takes $O(n^3)$ time to update \mathbf{Q} and \mathbf{C} because of computing the pseudo-inverse of an $n \times n$ matrix respectively. Fourthly, it knows easily that the time complexity for updating \mathbf{E}_2 is $O(n^2)$. Thus we can know that the computational burden of Algorithm 1 takes $O(n^3)$ totally in each iteration. Suppose the number of iterations is N , the total complexity of Algorithm 1 should be $N \times O(n^3)$.

4.2 Relationships Among LSMR and Some Related Algorithms

4.2.1 Is LSMR a Two-Step SMR?

Based on the deduction process presented in Sect. 3.1, researchers may consider that the results of LSMR could be easily achieved from the two steps: firstly, using SSC on the original data set \mathbf{X} to get \mathbf{Z} ; secondly, performing SMR on \mathbf{Z} to obtain \mathbf{C} . For alleviating this concern, we rewrite the objective function of LSMR (namely, Problem (4)) as follows:

$$\begin{aligned} & \min_{\mathbf{Z}, \mathbf{C}, \mathbf{E}_1, \mathbf{E}_2} \|\mathbf{Z}\|_1 + \alpha \|\mathbf{E}_1\|_1 + tr(\mathbf{C}\mathbf{L}_Z\mathbf{C}^t) + \beta \|\mathbf{Z} - \mathbf{Z}\mathbf{C}\|_F^2 \\ & = \min_{\mathbf{Z}, \mathbf{C}, \mathbf{E}_1, \mathbf{E}_2} \|\mathbf{Z}\|_1 + \alpha \|\mathbf{E}_1\|_1 + tr(\mathbf{C}\mathbf{L}_Z\mathbf{C}^t) + \beta tr(\mathbf{Z}(\mathbf{I}_n - \mathbf{C})(\mathbf{I}_n - \mathbf{C})^t\mathbf{Z}^t) \quad (20) \\ & = \min_{\mathbf{C}, \mathbf{Z}, \mathbf{E}_1, \mathbf{E}_2} \|\mathbf{Z}\|_1 + \alpha \|\mathbf{E}_1\|_1 + \beta tr(\mathbf{Z}\mathbf{L}_C\mathbf{Z}^t) + tr(\mathbf{C}\mathbf{L}_Z\mathbf{C}^t), \end{aligned}$$

where $\mathbf{L}_C = (\mathbf{I}_n - \mathbf{C})(\mathbf{I}_n - \mathbf{C})^t$. Then it can be seen that when \mathbf{C} is fixed, the above problem is actually a graph regularized SSC problem and $tr(\mathbf{Z}\mathbf{L}_C\mathbf{Z}^t)$ is the graph regularizer. This kind of graph regularizer is devised in the similar way used in [35] and has many excellent characters. Hence, LSMR could be regarded as an adaptive graph-regularized SSC.

4.2.2 The Relationship Between LSMR and GLRR

The objective function of graph-regularized low-rank representation (GLRR) [20] can be expressed as follows:

$$\begin{aligned} & \min_{\mathbf{Z}, \mathbf{E}} \|\mathbf{Z}\|_* + \beta tr(\mathbf{Z}\mathbf{L}_X\mathbf{Z}^t) + \alpha \|\mathbf{E}\|_{2,1}, \\ & s.t. \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \end{aligned} \quad (21)$$

where $\mathbf{L}_X = \mathbf{D}_X - \mathbf{W}_X$ is the Laplacian matrix which is designed in a similar way applied in SMR.

Compared the variant objective function of LSMR [namely, Eq. (20)] with GLRR, we can see: (1) similar to the existing spectral clustering methods, GLRR is performed in the original observed space; (2) the definitions of the Laplacian regularizer of LSMR and GLRR are different. Because of the Laplacian matrix in GLRR defined as same as that in SMR, it also has the same disadvantage, i.e. sensitive to the neighborhood size k ; (3) There are three parameters including the neighborhood size k and α, β needed to be adjusted in GLRR, which will make GLRR more difficult to get satisfied results. However, there are

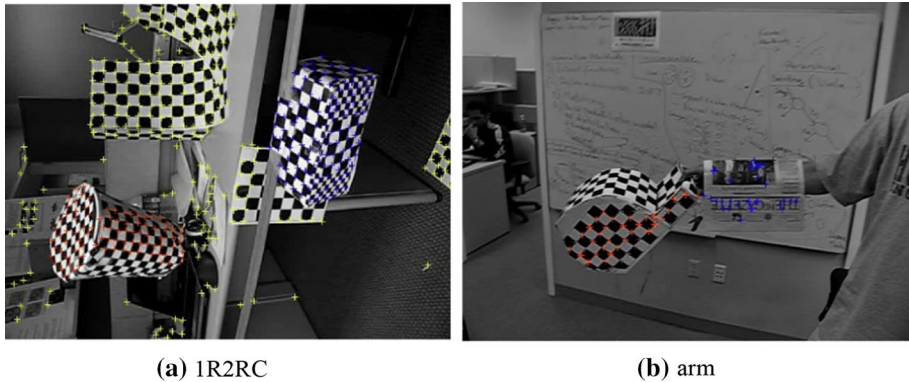


Fig. 2 Sample images of Hopkins 155 motion segmentation database

only two parameters including α and β in LSMR. Therefore, we can see that the performance of LSMR is superior to GLRR in our following experiments.

5 Experiments

In this section, plenty of subspace clustering experiments will be performed to verify the effectiveness of LSMR. For comparisons, some related algorithms including SSC [4], LRR [5], SMR [23], GLRR [20] will be also evaluated. Two types of databases will be adopted including Hopkins 155 motion segmentation database [36] and image databases (such as ORL database [37], the extended Yale B database [38], COIL-20 database [39] and MNIST (<https://yann.lecun.com/exdb/mnist/>)).

5.1 Experiments on Hopkins 155 Database

Hopkins 155 database [36] is a famous motion segmentation database which consists of 35 sequences of three motions³ and 120 sequences of two motions. The features of each sequence are extracted and tracked across all frames, furthermore errors were manually removed for each sequence. Subsequently each sequence could be taken as a single clustering task, consequently there are totally 155 clustering tasks. we use principal component analysis (PCA) [33] to project the data to be a 12-dimensional⁴ subspace. Figure 2 shows two sample images of Hopkins 155 database.

5.1.1 Sensitivity Analysis

We firstly tested the sensitivity of LSMR with different values of parameters α and β on the sub-database: “arm”. We let parameters α and β both vary in the interval [0.1,10]. The

³ It also contains a sequence of 5 motions which is called “dancing”. We neglect this sub-database in our experiments.

⁴ The choices of PCA dimension is followed the suggestion in [5].

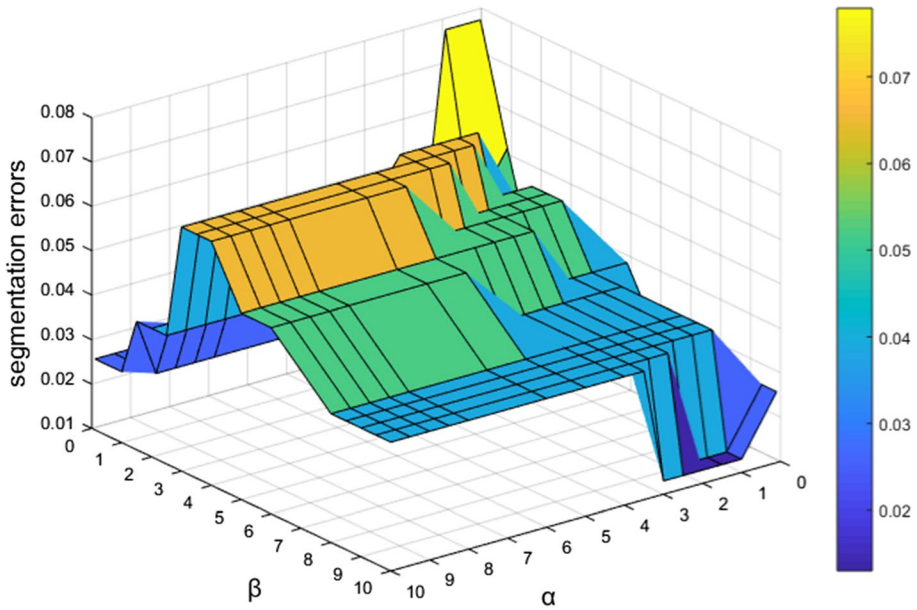


Fig. 3 The segmentation errors obtained by LSMR with different values of α and β on arm

segmentation errors⁵ obtained by LSMR with corresponding parameters are displayed in Fig. 3. Obviously, it could be discovered that the performance of LSMR is stable when α and β vary over relative large ranges.

5.1.2 Comparison with Related Algorithms

Moreover, for each evaluated algorithm, we also reported the detailed statistics of the segmentation errors including Mean, standard deviation (Std.) and maximal error (Max.) in Table 1. According to our previous sensitivity experiments, we set the two parameters in LSMR as $\alpha = 10$, $\beta = 1$. The parameters in other related algorithms were set to their best values by following the disciplines described in the corresponding related references. From Table 1, we can know that (1) the mean and maximal error of segmentation errors gotten by LSMR are all superior to other related algorithms; (2) the standard deviations gotten by LSMR on 2 motions and all data sets are all better than other related algorithms; (3) all the best values are obtained by LSMR or GLRR. The segmentation errors obtained by LSMR on some sequences of 3 motions are very small or zero. However, some results on the other sequences of 3 motions are a little bigger than those of GLRR. In other words, the segmentation errors obtained by LSMR on 3 motions fluctuate more widely than those of GLRR. Therefore, the standard deviation of our proposed method is not the best on 3 motions.

⁵ Segmentation error = 1 – segmentation accuracy.

Table 1 The segmentation errors (%) of different algorithms on Hopkins 155 database

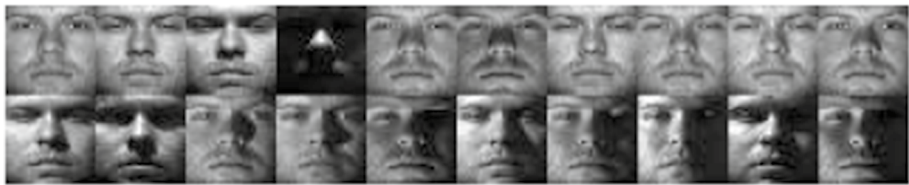
Method	2 motions			3 motions			ALL		
	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max
SSC	4.87	8.94	33.09	11.24	9.54	27.62	6.31	9.44	33.09
LRR	4.33	9.82	48.61	8.15	10.23	38.30	5.20	10.01	48.61
SMR	2.41	5.76	30.41	8.40	8.44	31.70	3.76	6.90	31.70
GLRR	2.36	5.45	33.09	7.93	8.22	31.70	3.62	6.58	33.09
LSMR	2.09	5.08	24.44	7.87	8.46	26.03	3.39	6.45	26.03

The optimal values of different criterion are emphasized in bold style

In SSC, $\lambda = 3$. In LRR, $\lambda = 2.4$. In SMR, $\lambda = 0.8, k = 4$. In GLRR, $\alpha = 0.6, \beta = 0.05, k = 4$



(a) sample images from ORL database



(b) sample images from the extended Yale B database

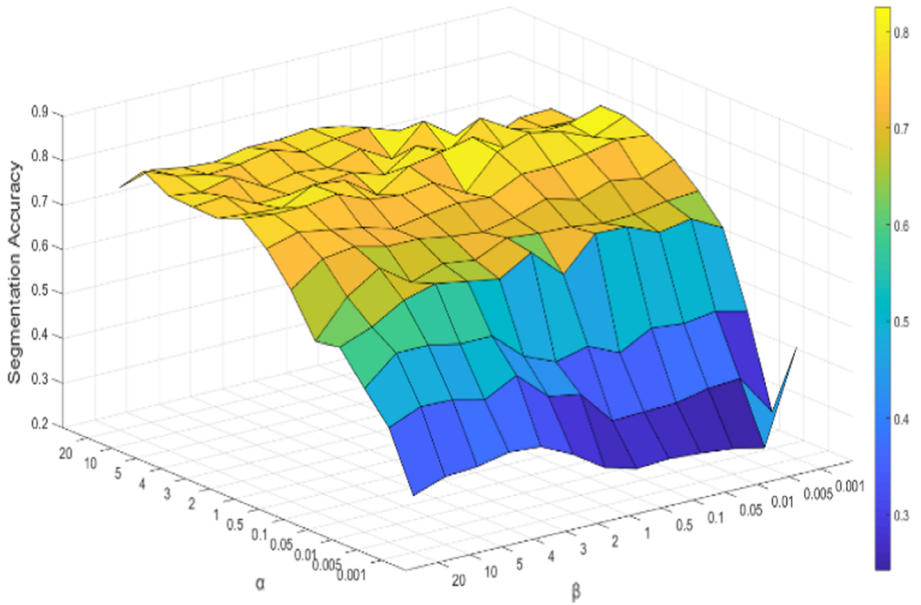
Fig. 4 Sample images from ORL and the extended Yale B databases

5.2 Experiments on Face Image Databases

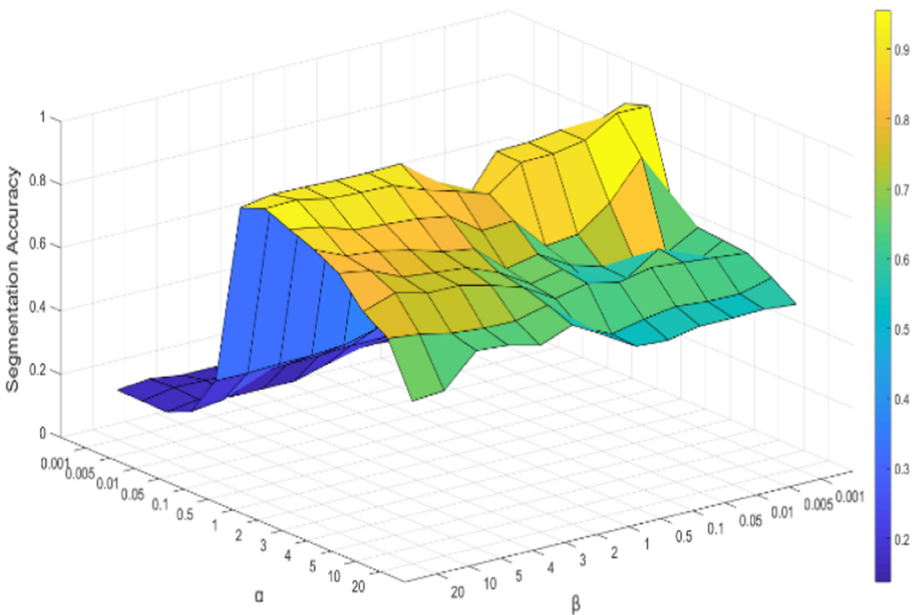
Two famous face image databases including ORL [37] and the extended Yale B [38] databases will be used in this subsection. The brief introductions of the two databases are as follows:

ORL database consists of 400 face images (without noise) of 40 persons that each individual has 10 different images. These images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses).

The extended Yale B face database consists of 38 human faces and around 64 near frontal images under different illuminations per person. Some images in this database were corrupted by shadow. We only chose face images from first 10 classes of the extended Yale B database to constitute a set of heavily corrupted data samples which contained totally 640 face images. In our experiments, all the images from ORL and the extended Yale B databases were resized to 32×32 pixels. What's more, the element value of each image vector was normalized into the interval $[0,1]$ by being divided 255 for effective computation. Some sample images from the two databases are respectively presented in Fig. 4a, b.



(a) ORL



(b) the extended Yale B

Fig. 5 The segmentation accuracies obtained by LSMR with different values of parameters on two face image databases

Table 2 The highest segmentation accuracies (%) obtained by different algorithms with their corresponding parameters on ORL and the extended Yale B databases

Database	Algorithm				
	SSC	LRR	SMR	GLRR	LSMR
ORL parameters	0.7675 $\lambda = 3$	0.7475 $\lambda = 3$	0.7750 $\lambda = 4$ $k = 4$	0.7925 $\alpha = 0.5$ $\beta = 2$ $k = 4$	0.8250 $\alpha = 2$ $\beta = 0.05$
Yale B parameters	0.8750 $\lambda = 0.5$	0.7016 $\lambda = 1$	0.6719 $\lambda = 8$ $k = 4$	0.6969 $\alpha = 0.05$ $\beta = 0.005$ $k = 4$	0.9547 $\alpha = 1$ $\beta = 0.005$

The optimal values of different criteria are emphasized in bold

5.2.1 Sensitivity Analysis

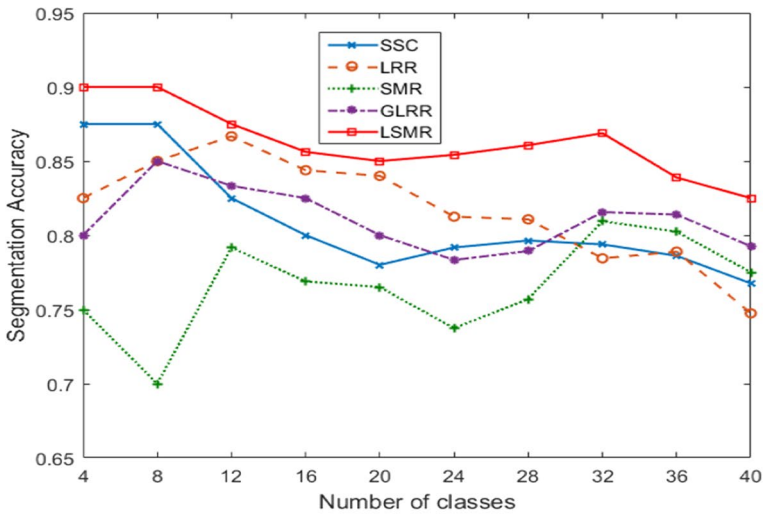
We used the two databases to test the sensitivity of LSMR with its two parameters as well. We chose face images from all 40 persons of ORL database and the first 10 persons of the extended Yale B database in this experiment. The parameters α and β were varied in relative large set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2, 3, 4, 5, 10, 20\}$. Figure 5 presents the segmentation accuracies obtained by LSMR with different values of two parameters, which demonstrates that LSMR is stable on a large range of α and β .

5.2.2 Comparison with Related Algorithms

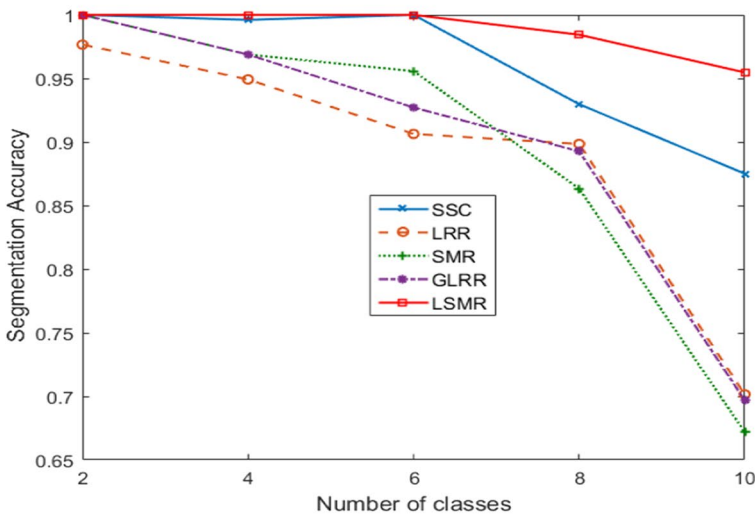
We also ran the other evaluated algorithms on the two databases. The best segmentation accuracies achieved by the five algorithms with their corresponding parameters are recorded in Table 2. From Table 2, we can know that (1) the highest segmentation accuracies on the two databases are all obtained by LSMR; (2) especially, the experimental result of LSMR is much better than that of the other evaluated algorithms on the extend Yale B database.

For further comparison, we carried out the subspace clustering experiments on some sub-databases chose from ORL and the extended Yale B databases. Each sub-database consisted of face images from q persons, where q changed from 4 to 40 in ORL database and 2 to 10 in the extended Yale B database respectively. After that, we performed the five evaluated algorithms to compute the subspace segmentation accuracies. In these experiments, we made the corresponding parameters in all evaluated algorithms vary in the interval $[0.0001, 20]$. Then we selected the best ones corresponding to the highest accuracies achieved by each evaluated algorithm. At last, the segmentation accuracy of each algorithm versus the number of class q are plotted in Fig. 6 on the two face image databases.

Obviously, from Fig. 6, we know that: (1) the best results are all obtained by LSMR in these experiments; (2) the performance of LSMR is much better than those of the other evaluated algorithms.



(a) ORL



(b) the extended Yale B

Fig. 6 The segmentation accuracies obtained by the evaluated algorithms versus the number of persons on ORL and the extended Yale B databases

5.3 Experiments on Other Image Databases

In addition, we expect to further verify high-performance of LSMR on different kinds of databases. Consequently, we adopted COIL-20 [39] and MNIST (<https://yann.lecun.com/exdb/mnist/>) in this subsection. The brief information of the two databases are introduced as the following:



(a) sample images from COIL-20 database

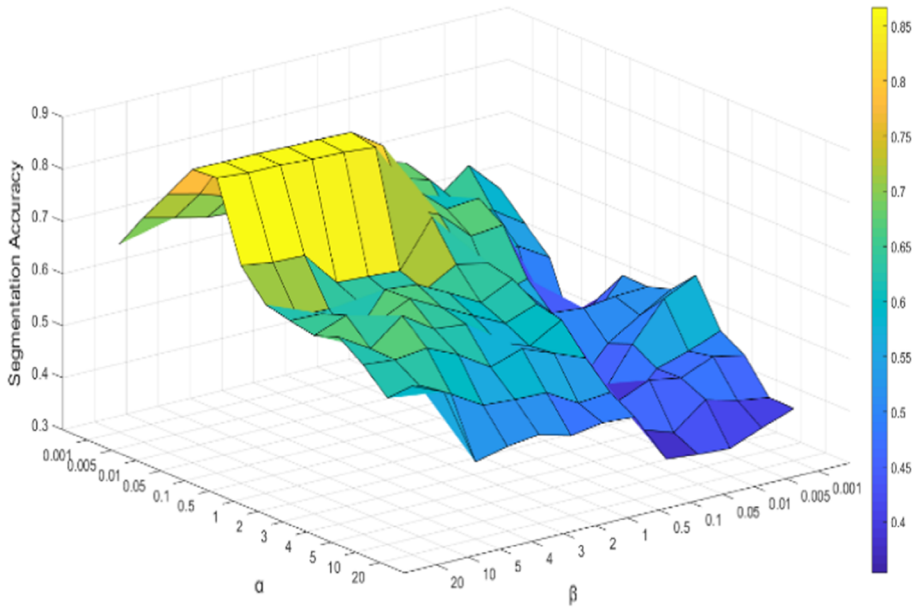


(b) sample images from MNIST database

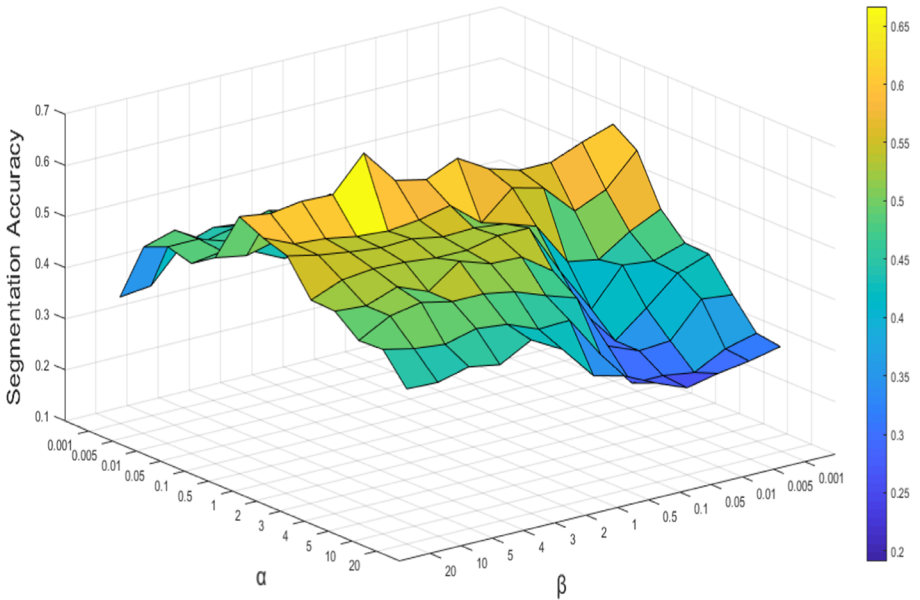
Fig. 7 Sample images from COIL-20 and MNIST database

COIL-20 database consists of 1440 images from 20 different subjects that each subject has 72 images. We formed a subset which consisted of first 36 images of each subject and images from the first 15 subjects were used. In our experiments, each image was resized to 32×32 pixels.

The MNIST database of handwritten digits has a training set of 60,000 examples and a test set of 10,000 examples. It has 10 subjects, corresponding to 10 handwritten digits, namely 0–9. In our experiments, we constructed a subset which consisted of the first 50 samples of each digit's training data set and resized each image to 28×28 pixels. Furthermore, the element value of each image vector was normalized into the interval $[0,1]$ by



(a) COIL-20



(b) MNIST

Fig. 8 The segmentation accuracies obtained by LSMR with different values of parameters on COIL-20 and MNIST databases

Table 3 The highest segmentation accuracies (%) obtained by different algorithms with their corresponding parameters on COIL-20 and MNIST databases

Database	Algorithm				
	SSC	LRR	SMR	GLRR	LSMR
COIL-20 parameters	0.7037 $\lambda = 0.05$	0.7333 $\lambda = 0.01$	0.7889 $\lambda = 0.005$ $k = 4$	0.8056 $\alpha = 0.001$ $\beta = 20$ $k = 4$	0.8667 $\alpha = 0.1$ $\beta = 20$
MNIST parameters	0.6200 $\lambda = 0.5$	0.5733 $\lambda = 1$	0.6422 $\lambda = 0.5$ $k = 4$	0.6467 $\alpha = 1$ $\beta = 20$ $k = 4$	0.6867 $\alpha = 0.3$ $\beta = 9.5$

The optimal values of different criteria are emphasized in bold

being divided 255 for effective computation. Figure 7a, b respectively show some sample images of the two databases.

5.3.1 Sensitivity Analysis

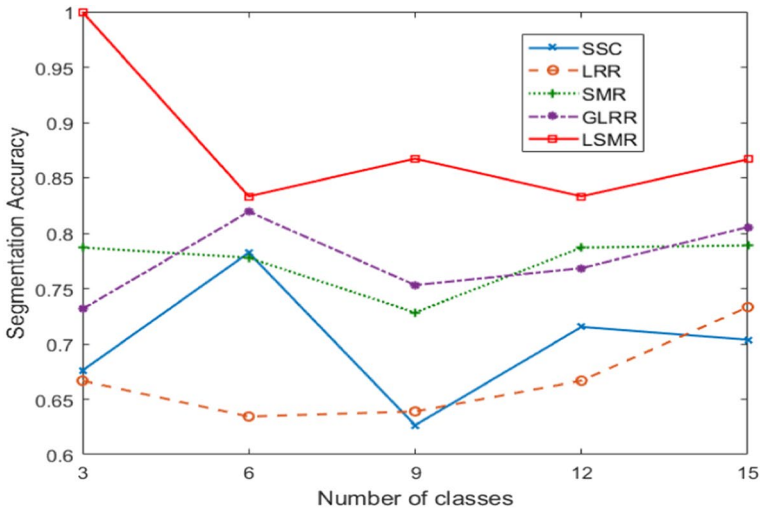
We used the methodology same as the previous experiments. Namely, we firstly tested the sensitivity of LSMR with different values of two parameters on COIL-20 and MNIST database. We selected images from the first 15 subjects of COIL-20 database and all 10 handwritten digits of MNIST database in this experiment. We made α and β both vary in the relative large set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2, 3, 4, 5, 10, 20\}$. The segmentation accuracy achieved by LSMR with corresponding parameters are illustrated in Fig. 8. Figure 8 clearly presents that LSMR isn't sensitive to the two parameters on the two databases.

5.3.2 Comparison with Related Algorithms

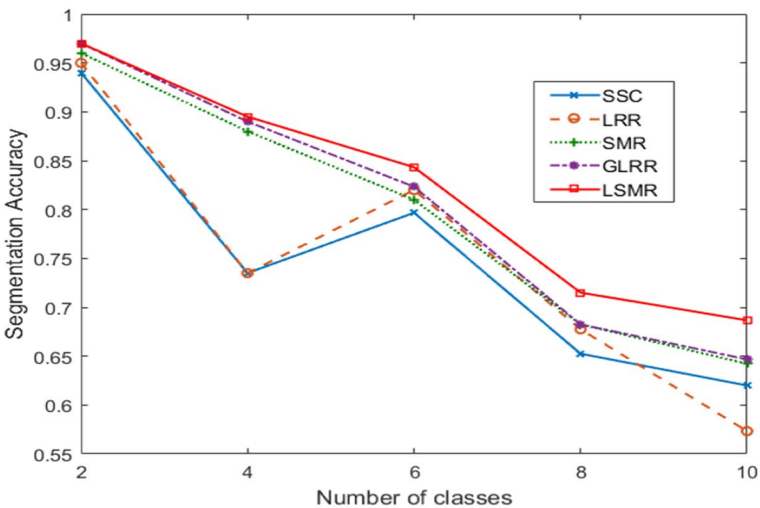
After all, each of the evaluated algorithms was conducted on chosen image samples from COIL-20 and MNIST databases. Table 3 also records the high segmentation accuracies achieved by all of the evaluated algorithms with their corresponding parameters. From Table 3, it could be seen that the highest segmentation accuracies on COIL-20 and MNIST databases are all obtained by LSMR. Hence, LSMR outperforms other evaluated algorithms.

Then subspace clustering experiments were also conducted on some sub-databases constructed from COIL-20 and MNIST databases. Each sub-database consisted of image samples from q subjects, where q changed from 3 to 15 in COIL-20 database and 2 to 10 in MNIST database. Subsequently the five evaluated algorithms were performed to obtain subspace segmentation accuracies. The corresponding parameters in all of evaluated algorithms were varied in the interval $[0.0001, 20]$. We selected the best values corresponding to the highest accuracies of all of evaluated algorithms. Figure 9 plotted the segmentation accuracy curve of each algorithm vs. the number of class q .

Obviously, from Fig. 9, we could find that: (1) the best results are all achieved by LSMR in these experiments; (2) the results of LSMR are much better than those of the other related algorithms in the two experiments. According to all the experiments, we can conclude that LSMR is an efficient algorithm for subspace clustering.



(a) COIL-20



(b) MNIST

Fig. 9 The segmentation accuracies obtained by the evaluated algorithms versus the number of subjects on COIL-20 and MNIST databases

6 Conclusion

In this paper, we devised a new subspace clustering algorithm, called latent smooth representation clustering (LSMR), in which we regarded sparse subspace clustering (SSC) found a latent subspace for the original data set and smooth representation clustering (SMR) was consequently applied in the latent space to handle subspace clustering tasks. We discussed the relationships between LSMR and some existing related algorithms. Finally, subspace

clustering experiments conducted on Hopkins 155 database and four image databases proved LSMR was superior to some existing related algorithms.

Compliance with Ethical Standards

Conflict of interest The authors declared that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. SIGKDD Explor Newsl ACM Spec Interes Gr Knowl Discov Data Min 6:90
2. Vidal R (2011) Subspace clustering. *IEEE Signal Process Mag* 28:52–68. <https://doi.org/10.1109/MSP.2010.939739>
3. Elhamifar E, Vidal R (2009) Sparse subspace clustering. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop CVPR Workshops 2009, IEEE, pp 2790–2797. <https://doi.org/10.1109/CVPRW.2009.5206547>
4. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell* 35:2765–2781. <https://doi.org/10.1109/TPAMI.2013.57>
5. Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: ICML 2010—proceedings, 27th international conference on machine learning
6. Liu G, Lin Z, Yan S et al (2013) Robust recovery of subspace structures by low-rank representation. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2012.88>
7. Rao S, Tron R, Vidal R, Ma Y (2010) Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2009.191>
8. Ma Y, Derksen H, Hong W, Wright J (2007) Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2007.1085>
9. Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). *Pattern Recognit Lett*. <https://doi.org/10.1016/j.patrec.2013.08.006>
10. Wei L, Wu A, Yin J (2015) Latent space robust subspace segmentation based on low-rank and locality constraints. *Expert Syst Appl* 42:6598–6608. <https://doi.org/10.1016/j.eswa.2015.04.041>
11. Shi J, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/34.868688>
12. Liu R, Lin Z, De La Torre F, Su Z (2012) Fixed-rank representation for unsupervised visual learning. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
13. Liu G, Yan S (2011) Latent low-rank representation for subspace segmentation and feature extraction. In: Proceedings of the IEEE international conference on computer vision
14. Li CG, Vidal R (2015) Structured sparse subspace clustering: a unified optimization framework. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
15. Li CG, You C, Vidal R (2017) Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. *IEEE Trans Image Process* 26:2988–3001. <https://doi.org/10.1109/TIP.2017.2691557>
16. Chen H, Wang W, Feng X (2018) Structured sparse subspace clustering with within-cluster grouping. *Pattern Recognit*. <https://doi.org/10.1016/j.patcog.2018.05.020>
17. Zhuang L, Gao H, Lin Z et al (2012) Non-negative low rank and sparse graph for semi-supervised learning. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
18. Tang K, Liu R, Su Z, Zhang J (2014) Structure-constrained low-rank representation. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2014.2306063>

19. Wei L, Wang X, Yin J, Wu A (2016) Spectral clustering steered low-rank representation for subspace segmentation. *J Vis Commun Image Represent* 38:386–395. <https://doi.org/10.1016/j.jvcir.2016.03.017>
20. Lu X, Wang Y, Yuan Y (2013) Graph-regularized low-rank representation for destriping of hyperspectral images. *IEEE Trans Geosci Remote Sens* 51:4009–4018. <https://doi.org/10.1109/TGRS.2012.2226730>
21. Yin M, Gao J, Lin Z (2016) Laplacian regularized low-rank representation and its applications. *IEEE Trans Pattern Anal Mach Intell* 38:504–517. <https://doi.org/10.1109/TPAMI.2015.2462360>
22. Lu CY, Min H, Zhao ZQ, et al (2012) Robust and efficient subspace segmentation via least squares regression. Lecture notes in computer science (including subseries lecture notes in artificial intelligence lecture notes in bioinformatics). LNCS, vol 7578, pp 347–360. https://doi.org/10.1007/978-3-642-33786-4_26
23. Hu H, Lin Z, Feng J, Zhou J (2014) Smooth representation clustering. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. <https://doi.org/10.1109/CVPR.2014.484>
24. Yu J, Tao D, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2014.2336697>
25. Hong C, Yu J, Tao D, Wang M (2015) Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. *IEEE Trans Ind Electron* 62:3742–3751. <https://doi.org/10.1109/TIE.2014.2378735>
26. Hong C, Yu J, Zhang J et al (2019) Multimodal face-pose estimation with multitask manifold deep learning. *IEEE Trans Ind Inform*. <https://doi.org/10.1109/TII.2018.2884211>
27. Zhang J, Yu J, Tao D (2018) Local deep-feature alignment for unsupervised dimension reduction. *IEEE Trans Image Process*. <https://doi.org/10.1109/TIP.2018.2804218>
28. Yu J, Tan M, Zhang H et al (2019) Hierarchical deep click feature prediction for fine-grained image recognition. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/tpami.2019.2932058>
29. Yu J, Zhu C, Zhang J et al (2020) Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2019.2908982>
30. Yu J, Yao J, Zhang J et al (2020) SPRNet: single-pixel reconstruction for one-stage instance segmentation. *IEEE Trans Cybern*. <https://doi.org/10.1109/tcyb.2020.2969046>
31. Lin Z, Chen M, Ma Y (2010) The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. <https://doi.org/10.1016/j.jsb.2012.10.010>
32. Wright J, Yang AY, Ganesh A et al (2009) Robust face recognition via sparse representation. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2008.79>
33. Duda RO, Hart PE, Stork DG (2001) *Pattern classification*. Wiley, New York
34. Bartels RH, Stewart GW (1972) Solution of the matrix equation $AX + XB = C$ [F4]. *Commun ACM*. <https://doi.org/10.1145/361573.361582>
35. Wei L, Wang X, Wu A et al (2018) Robust subspace segmentation by self-representation constrained low-rank representation. *Neural Process Lett* 48:1671–1691. <https://doi.org/10.1007/s11063-018-9783-y>
36. Tron R, Vidal R (2007) A benchmark for the comparison of 3-D motion segmentation algorithms. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*
37. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: *IEEE workshop on applications of computer vision—proceedings*
38. Lee KC, Ho J, Kriegman DJ (2005) Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/TPAMI.2005.92>
39. Nene S, Nayar S, Murase H (1996) *Columbia object image library (COIL-20)*. Tech Rep

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.