



Graph constraint-based robust latent space low-rank and sparse subspace clustering

Yunjun Xiao¹ · Jia Wei¹ · Jiabing Wang¹ · Qianli Ma¹ · Shandian Zhe² · Tolga Tasdizen³

Received: 10 February 2019 / Accepted: 18 June 2019 / Published online: 27 June 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Recently, low-rank and sparse representation-based methods have achieved great success in subspace clustering, which aims to cluster data lying in a union of subspaces. However, most methods fail if the data samples are corrupted by noise and outliers. To solve this problem, we propose a novel robust method that uses the F-norm for dealing with universal noise and the l_1 norm or the $l_{2,1}$ norm for capturing outliers. The proposed method can find a low-dimensional latent space and a low-rank and sparse representation simultaneously. To preserve the local manifold structure of the data, we have adopted a graph constraint in our model to obtain a discriminative latent space. Extensive experiments on several face benchmark datasets show that our proposed method performs better than state-of-the-art subspace clustering methods.

Keywords Dimension reduction · Low-rank and sparse representation · Subspace clustering · Manifold clustering

1 Introduction

In the field of computer vision and pattern recognition, dealing with high-dimensional data is a challenging task. In most cases, these high-dimensional data can be represented by a low-dimensional subspace. For example, the face data under different illumination conditions can be

approximated by a low-dimensional linear subspace [1]. Subspace clustering aims to cluster the data points drawn from a union of low-dimensional subspaces.

The subspace clustering problem has attracted attention in recent years, and many methods have been proposed to address it, including iterative methods [2–4], algebraic methods [5–7], statistical methods [8–11], and methods based on spectral clustering [12–22]. Researchers favor the methods based on spectral clustering, which can obtain better results. These methods divide the spectral clustering problem into two steps. First, an affinity matrix is built from the data. Then, spectral clustering is applied on the affinity matrix to segment the data. The basis of such methods is to construct an informative affinity matrix.

Recently, various algorithms based on sparse representation [13, 14] or low-rank representation [15–17, 20, 22] have been proposed in the literature for subspace clustering. Such methods are robust to noise and outliers and do not require the knowledge of the dimensions and the number of subspaces. Most of these methods are based on a self-expressiveness model, which states that each sample can be expressed as a linear combination of other samples. They utilize all samples as a dictionary and find a sparse or low-rank representation matrix. With relatively clean data, $\mathbf{X} = \mathbf{XR}$, where \mathbf{X} is the data matrix and \mathbf{R} is the coefficient matrix. With corrupted data, the constraint is changed

✉ Jia Wei
csjwei@scut.edu.cn

Yunjun Xiao
csxyunjun@mail.scut.edu.cn

Jiabing Wang
jbwang@scut.edu.cn

Qianli Ma
qianlima@scut.edu.cn

Shandian Zhe
zhe@cs.utah.edu

Tolga Tasdizen
tolga@sci.utah.edu

¹ School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

² School of Computing, University of Utah, Salt Lake City, USA

³ Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, USA

to $\mathbf{X} = \mathbf{X}\mathbf{R} + \mathbf{E}$, where \mathbf{E} denotes the reconstruction error. In general, the subspace clustering problem can be summed up as the following optimization problem:

$$\min_{\mathbf{R}, \mathbf{E}} \|\mathbf{R}\|_k + \lambda \|\mathbf{E}\|_l \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{R} + \mathbf{E}, \text{diag}(\mathbf{R}) = \mathbf{0} \quad (1)$$

where $\|\cdot\|_k$ and $\|\cdot\|_l$ are suitable norms, λ is a trade-off parameter, and the constraint $\text{diag}(\mathbf{R}) = \mathbf{0}$ is optionally used to avoid a trivial solution of \mathbf{R} .

The methods differ in the choice of norms for the regularization on \mathbf{R} and/or the reconstruction error \mathbf{E} . For example, Elhamifar and Vidal proposed sparse subspace clustering (SSC) [13, 14] based on sparse representation (SR) [23]. The l_1 norm is chosen for $\|\cdot\|_k$ to get a relatively sparse matrix \mathbf{R} , and the F-norm and/or the l_1 norm of \mathbf{E} are used to handle Gaussian noise and/or outliers. Liu et al. [16, 17] argued that due to learning the sparse representation of each data vector individually, SSC may not be able to capture the global structure of data, and the performance of SSC will be largely depressed when the data is grossly corrupted. Consequently, they proposed a method called the low-rank representation (LRR), which finds the lowest rank representation of all data jointly. They used the nuclear norm $\|\cdot\|_*$ for $\|\cdot\|_k$ to obtain a relatively low-rank matrix \mathbf{R} and the $l_{2,1}$ norm of \mathbf{E} to tackle sample-specific corruptions, where some data vectors are corrupted and others are clean. Vidal et al. [20] held that using data as the dictionary is unreasonable when data are grossly corrupted. They learned a noise-free dictionary from corrupted data and found the low-rank representation based on the learned dictionary. They named this algorithm low-rank subspace clustering (LRSC). In LRSC, the nuclear norm $\|\cdot\|_*$ is applied to $\|\cdot\|_k$, the F-norm, and the l_1 norm of \mathbf{E} is used to handle Gaussian noise and outliers. Wang et al. [21] reported that the representation matrix is often simultaneously sparse and low rank, and they therefore combined SSC and LRR, termed low-rank sparse subspace clustering (LRSSC). In LRSSC, the nuclear norm $\|\cdot\|_*$ and the $\|\cdot\|_1$ norm are used for $\|\cdot\|_k$, and the F-norm of \mathbf{E} is used to deal with Gaussian noise. Also, several variants of these algorithms have been proposed, such as least squares regression (LSR) [19], l_0 -SSC [24], SSC-learned orthogonal matching pursuit (SSC-LOMP) [25], subspace clustering with learning an adaptive low-rank graph (SC-LALRG) [26], and subspace clustering with block diagonal representation (BDR) [27]. These methods have yielded state-of-the-art results in different subspace segmentation tasks.

However, finding sparse or low-rank representation is time-consuming when the dimension of the feature is high [14]. To tackle this problem, we can process the high-dimensional data into low-dimensional data using principal component analysis [28] or random projections [29] prior to

obtaining sparse or low-rank representation. The major drawback of this method is that it may lose too much information, leading to poor performance. A more reasonable approach is to find the low-dimensional embedding and sparse or low-rank representation jointly. Patel et al. [30] proposed an extension of SSC: latent space sparse subspace clustering (LS3C). Given a dataset, LS3C finds a low-dimensional embedding and a sparse representation matrix simultaneously. They also proposed an extension of LS3C: latent space low-rank and sparse subspace clustering (LSLRSSC) [31]. LSLRSSC finds a low-dimensional embedding and a sparse and low-rank representation matrix simultaneously. Also, Wei et al. [32] proposed an extension LS3C, which they called latent space robust subspace segmentation (LSRS2). In LSRS2, the weighted l_1 norm and the nuclear norm are used for the representation matrix. These methods learn a low-dimensional space without losing too much information, and a PCA-like regularization term is added to their objective functions. PCA aims to discover the global structure of the Euclidean space; nevertheless, the local manifold structure is more necessary than the global Euclidean structure in many real-world application problems [33, 34]. Tang et al. [35] designed a special term describing the property of learned space instead of PCA that can keep the local manifold structure of the high-dimensional data: robust subspace learning-based low-rank representation (RSLRR). RSLRR learns a low-rank representation matrix, but the representation matrix is often sparse and low rank. In addition, the methods mentioned above using the F-norm or the $l_{2,1}$ norm to capture the reconstruction error are not robust [36–38].

Consequently, in this paper, we propose a novel algorithm, robust latent space low-rank and sparse subspace clustering (RLSLRSSC). To preserve the local manifold structure of the high-dimensional data, we devise a graph constraint term instead of a PCA-like term in the objective function of our proposed algorithm. Moreover, to make our algorithm robust to noise, we characterize the reconstruction error more precisely. Due to the limited representational capability with insufficient data and the error in data acquisition and transmission, we assume a portion of the error obeys a Gaussian distribution. Hence, we use the F-norm to depict Gaussian noise. We use the l_1 norm or the $l_{2,1}$ norm to depict outliers. In summary, our main contributions are as follows:

1. By adopting a graph constraint term rather than a PCA-like term, we propose a novel representation learning method that finds a discriminating latent space and a low-rank and sparse representation matrix simultaneously.
2. We depict the noise more accurately, so the robustness of our algorithm will be enhanced.

3. A simple and efficient iterative method is proposed to optimize our algorithm.

The remainder of this paper is organized as follows. In Sect. 2, we briefly review some related work. Section 3 presents the proposed method RLSLRSSC. In Sect. 4, we describe the optimization algorithms, the time complexity, and convergence analysis. Section 5 reports experimental results. Section 6 concludes this paper.

2 Related work

In this section, we briefly review sparse and low-rank subspace clustering methods, including SSC [13, 14], LRR [16, 17], LRSSC [21], and LSLRSSC [31]. For brevity, we summarize some notations in Table 1.

2.1 Sparse subspace clustering (SSC)

Sparse representation (SR) [23] is an effective tool for representing and compressing high-dimensional data. In the recent years, an increasing number of methods have been proposed based on SR. Also, a lot of applications derive from SR, such as Tree2Vector [39, 40]. Similarly, SSC [13, 14] learns the affinity matrix for a dataset based on SR. Given a dataset, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbf{R}^{D \times N}$ drawn from a union of Q independent linear subspaces S_1, S_2, \dots, S_Q . Each sample $\mathbf{x}_i \in \mathbf{R}^D$ can be linearly and approximately represented by a combination of a few points from the rest of the samples in \mathbf{X} . The idea of SSC can be expressed as follows:

$$\min_{\mathbf{R}} \|\mathbf{R}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{XR}, \text{diag}(\mathbf{R}) = \mathbf{0} \quad (2)$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N] \in \mathbf{R}^{N \times N}$ is a coefficient matrix whose column \mathbf{r}_i is the representation vector corresponding to \mathbf{x}_i , $\text{diag}(\mathbf{R}) \in \mathbf{R}^N$ is the vector whose elements equal the diagonal elements of \mathbf{R} , and $\mathbf{0} \in \mathbf{R}^N$ is a vector containing zeros only.

Table 1 Summary of main notations used in this paper

Matrix	Bold capital symbol, e.g., \mathbf{X}
Vector	Bold lowercase symbol, e.g., \mathbf{x}_i
\mathbf{X}^T	Transpose of matrix \mathbf{X}
\mathbf{W}_{ij}	The (i, j) th of matrix \mathbf{W}_{ij}
$\ \cdot\ _F$	F-norm, i.e., $\ \mathbf{M}\ _F = \sqrt{\sum_{i,j} \mathbf{M}_{i,j}^2}$
$\ \cdot\ _1$	l_1 -norm, i.e., $\ \mathbf{M}\ _1 = \sum_{i,j} \mathbf{M}_{i,j} $
$\ \cdot\ _{2,1}$	$l_{2,1}$ -norm, i.e., $\ \mathbf{M}\ _{2,1} = \sum_j \sqrt{\sum_i \mathbf{M}_{i,j}^2}$
$\ \cdot\ _*$	Nuclear norm, i.e., sum of the singular values
\mathbf{I}	Identity matrix

In fact, the data lie in a union of affine subspaces. In this condition, we obtain the following optimization:

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

where $\mathbf{1}$ is a vector containing one only. Considering that the data may be corrupted, the constraint $\mathbf{X} = \mathbf{XR}$ should be relaxed to $\mathbf{X} = \mathbf{XR} + \mathbf{E}$. \mathbf{E} is the noise matrix, and the F-norm is used to capture the noise. The following problem, therefore, can be solved to obtain the representation matrix \mathbf{R} :

$$\min_{\mathbf{R}} \|\mathbf{X} - \mathbf{XR}\|_F^2 + \lambda \|\mathbf{R}\|_1 \quad \text{s.t.} \quad \text{diag}(\mathbf{R}) = \mathbf{0}, \mathbf{R}^T \mathbf{1} = \mathbf{1} \quad (4)$$

where λ is a regularization parameter. This problem can be solved by using the alternating direction method of multipliers (ADMM) [41].

2.2 Low-rank representation (LRR)

LRR [16, 17] uses low-rank representation to construct an affinity matrix for a group of data. Similar to SSC, LRR uses the data \mathbf{X} as the dictionary. When the data is noise-free, a low-rank representation \mathbf{R} can be obtained by solving the problem

$$\min_{\mathbf{R}} \text{rank}(\mathbf{R}) \quad \text{s.t.} \quad \text{diag}(\mathbf{R}) = \mathbf{0} \quad (5)$$

However, the above optimization problem is difficult to solve. As a common approach to addressing the above problem, the rank function is replaced by its convex surrogate, the nuclear norm. When the data is corrupted by some arbitrary noise \mathbf{E} , i.e., $\mathbf{X} = \mathbf{XR} + \mathbf{E}$, the $l_{2,1}$ norm is employed to deal with the noise. The objective function of LRR is

$$\min_{\mathbf{R}, \mathbf{E}} \|\mathbf{R}\|_* + \lambda \|\mathbf{E}\|_{2,1} \quad \text{s.t.} \quad \mathbf{X} = \mathbf{XR} + \mathbf{E} \quad (6)$$

The above problem can be solved by an inexact augmented Lagrange multiplier (ALM) method [42].

2.3 Low-rank and sparse subspace clustering (LRSSC)

The representation matrix \mathbf{R} is often simultaneously sparse and low rank. By combining SSC and LRR, a low-rank and sparse representation can be learned. In LRSSC [21], the representation matrix can be found by solving the following optimization problem:

$$\min_{\mathbf{R}} \|\mathbf{X} - \mathbf{XR}\|_F^2 + \lambda_1 \|\mathbf{R}\|_* + \lambda_2 \|\mathbf{R}\|_1 \quad (7)$$

$$\text{s.t.} \quad \text{diag}(\mathbf{R}) = \mathbf{0}, \mathbf{R}^T \mathbf{1} = \mathbf{1}$$

where λ_1 and λ_2 are two regularization parameters. This problem can be solved using the ADMM [41] method.

2.4 Latent space low-rank and sparse subspace clustering (LSLRSSC)

Finding low-rank and sparse representation is very time-consuming. Dissimilar to LRSSC, LSLRSSC [31] finds a latent low-dimensional space and a low-rank and sparse representation matrix simultaneously. LSLRSSC can be expressed as the following problem:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{R}} \quad & \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R}\|_F^2 + \lambda_1 \|\mathbf{R}\|_* + \lambda_2 \|\mathbf{R}\|_1 \\ & + \lambda_3 \|\mathbf{X} - \mathbf{P}^T \mathbf{P}\mathbf{X}\|_F^2 \text{ s.t. } \text{diag}(\mathbf{R}) = \mathbf{0}, \mathbf{R}^T \mathbf{1} = \mathbf{1}, \mathbf{P}\mathbf{P}^T = \mathbf{I} \end{aligned} \quad (8)$$

where $\lambda_1, \lambda_2, \lambda_3$ are three trade-off parameters and \mathbf{P} is a projection matrix. The last term is used for guaranteeing that the projection does not lose too much information in the original domain. This problem can be solved by optimizing \mathbf{P} and \mathbf{R} alternatively using ADMM [41]. In SSC, LRR, LRSSC, and LSLRSSC, once the representation matrix \mathbf{R} is learned, the affinity matrix \mathbf{W} can be constructed by $\mathbf{W} = |\mathbf{R}| + |\mathbf{R}^T|$, where $|\mathbf{R}|$ is the modulus of \mathbf{R} . Finally, spectral clustering [43] is used to obtain clustering results.

3 Robust latent space low-rank and sparse subspace clustering (RLSLRSSC)

In this section, we present a novel method, robust latent space low-rank and sparse subspace clustering (RLSLRSSC). Our model consists of three steps. First, it projects the high-dimensional data onto a lower dimensional space, and we add a graph constraint to promote the discriminative ability of the projected space. Second, the projected data is represented by a low-rank and sparse representation matrix. Third, we need to deal with the noise and outliers. Simply, we aim to learn a low-rank and sparse representation matrix and a discriminative projection matrix simultaneously by using a graph constraint. RLSLRSSC can be expressed as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{R}, \mathbf{G}, \mathbf{E}} \quad & \lambda f(\mathbf{G}) + \lambda_1 f_1(\mathbf{R}) + \lambda_2 f_2(\mathbf{P}) + \lambda_3 f_3(\mathbf{E}) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{X}\mathbf{R} + \mathbf{G} + \mathbf{E}, \mathbf{P}\mathbf{P}^T = \mathbf{I}, \mathbf{R}^T \mathbf{1} = \mathbf{1} \end{aligned} \quad (9)$$

where $\lambda, \lambda_1, \lambda_2, \lambda_3$ are four regularization parameters, \mathbf{P} is a projection matrix, \mathbf{R} is a low-rank and sparse representation matrix, \mathbf{G} denotes universal noise, and \mathbf{E} denotes outliers. In most cases, universal noise obeys a Gaussian distribution, and we can apply the F-norm for \mathbf{G} , $f(\mathbf{G}) = \|\mathbf{G}\|_F^2$. One can find both sparse and low-rank representation as is performed in LRSSC by setting

$f_1(\mathbf{R}) = \|\mathbf{R}\|_* + \alpha \|\mathbf{R}\|_1$, where α is a trade-off parameter. To deal with outliers, the l_1 norm or the $l_{2,1}$ norm is employed for \mathbf{E} , $f_3(\mathbf{E}) = \|\mathbf{E}\|_1$ or $f_3(\mathbf{E}) = \|\mathbf{E}\|_{2,1}$. $f_2(\mathbf{P})$ describes the property of the learned space, which enforces discrimination of the learned space.

The graph constraint term $f_2(\mathbf{P})$ learns a transformation matrix, which can preserve useful information and projects the data onto a discriminative latent space, where different classes are separated more easily compared with the original space. Because the local manifold structure is more significant than the global Euclidean structure [33, 34], preserving the local manifold structure in the projected space is more reasonable. We hope that if two data points \mathbf{x}_i and \mathbf{x}_j are close in the original space, their corresponding low-dimensional embedding \mathbf{y}_i and \mathbf{y}_j should be close, too. $f_2(\mathbf{P})$ is designed to preserve the preferable relationship between the data samples. As a result, we choose the k nearest neighbors of each data point by Euclidean distance. Then, the adjacency graph \mathbf{W} can be obtained with the k nearest neighbors by using radial basis function (RBF). In RBF, it is important to select the value of σ . However, when the data includes clusters with different local statistics, the adjacency graph may not be accurate even using the optimal σ . In addition, it is hard to get the optimal σ because there are many parameters in our model. To overcome these problems, \mathbf{W} is defined as

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma_i \sigma_j}\right) & \mathbf{x}_i \in N(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $N(\mathbf{x}_i)$ denotes the nearest neighbors of \mathbf{x}_i and σ_i is a local scaling [44] parameter for \mathbf{x}_i , which permits self-tuning of the point-to-point distances according to the local statistics of the neighborhoods of \mathbf{x}_i . The local scale σ_i can be defined as $\sigma_i = d(\mathbf{x}_i, \mathbf{x}_t)$, where \mathbf{x}_t is the t th neighbor of \mathbf{x}_i .

We formulate the graph constraint term $f_2(\mathbf{P})$ as

$$\min_{\mathbf{P}} f_2(\mathbf{P}) = \sum_{i=1}^N \sum_{j=1}^N \frac{1}{2} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \mathbf{W}_{ij} \quad (11)$$

where \mathbf{y}_i is the low-dimensional embedding corresponding to \mathbf{x}_i and we have $\mathbf{y}_i = \mathbf{P}\mathbf{x}_i$. Let \mathbf{D} be a diagonal matrix with $\mathbf{D}_{jj} = \sum_{i=1}^N \mathbf{W}_{ij}$, and \mathbf{L} be the Laplacian matrix with $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The objective function (11) can be modified as the following equivalent problem:

$$\min_{\mathbf{P}} f_2(\mathbf{P}) = \text{tr}(\mathbf{P}\mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{P}^T) \text{ s.t. } \mathbf{P}\mathbf{P}^T = \mathbf{I} \quad (12)$$

The above optimization problem can be seen as a simpler one in orthogonal locality preserving projections (OLPP) [45–47]. If the sample size is less than the feature

dimension of the sample, we will come across the under-sampled size problem. To ensure that the matrix $\mathbf{M} = \mathbf{X}\mathbf{L}\mathbf{X}^T$ will be nonsingular, we may use PCA to reduce the dimensionality of the data in advance.

By expanding $f(\mathbf{G})$, $f_1(\mathbf{R})$, $f_2(\mathbf{P})$, and $f_3(\mathbf{E})$, our proposed model (9) can be expressed as

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{R}, \mathbf{E}} \quad & \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_1 \|\mathbf{R}\|_* + \lambda_2 \|\mathbf{R}\|_1 \\ & + \lambda_3 \text{tr}(\mathbf{P}\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}^T) + \lambda_4 \|\mathbf{E}\|_l \quad \text{s.t. } \mathbf{R}^T\mathbf{1} = \mathbf{1}, \mathbf{P}\mathbf{P}^T = \mathbf{I} \end{aligned} \tag{13}$$

where $\|\cdot\|_l$ denotes a certain norm. We consider two kinds of norms on \mathbf{E} : (i) the l_1 norm, resulting in a model denoted by RLSLRSSC-L1; and (ii) the $l_{2,1}$ norm, resulting in a model denoted by RLSLRSSC-L21.

4 Optimization

The optimization problem (13) is difficult to solve, and it is not jointly convex concerning $(\mathbf{P}, \mathbf{R}, \mathbf{E})$. We can divide the optimization problem (13) into three subproblems to find the projection matrix \mathbf{P} , the representation matrix \mathbf{R} , and the error matrix \mathbf{E} separately. These subproblems are solved alternatively by updating one variable and fixing the other ones. Each subproblem will be discussed in detail in the following section.

4.1 Update of projection matrix P

Given fixed \mathbf{R} and \mathbf{E} , the cost function in Eq. (13) is further reduced to

$$\begin{aligned} J_{\mathbf{P}} = \min_{\mathbf{P}} \quad & \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_3 \text{tr}(\mathbf{P}\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}^T) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I} \end{aligned} \tag{14}$$

The above objective function is nonconvex with a local minimum of it obtained as follows. First, considering $\mathbf{P}\mathbf{P}^T = \mathbf{I}$, we have

$$\|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 = \text{tr}(\mathbf{P}\Phi(\mathbf{P})\mathbf{P}^T) \tag{15}$$

where $\Phi(\mathbf{P}) = (\mathbf{X} - \mathbf{X}\mathbf{R} - \mathbf{P}^T\mathbf{E})(\mathbf{X} - \mathbf{X}\mathbf{R} - \mathbf{P}^T\mathbf{E})^T$. In this case, the objective function in Eq. (14) can then be rewritten as

$$\begin{aligned} J_{\mathbf{P}} = \min_{\mathbf{P}} \quad & \lambda \text{tr}(\mathbf{P}\Phi(\mathbf{P})\mathbf{P}^T) + \lambda_3 \text{tr}(\mathbf{P}\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}^T) \\ \min_{\mathbf{P}} \quad & \text{tr}(\mathbf{P}(\lambda\Phi(\mathbf{P}) + \lambda_3\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{P}^T) \\ \text{s.t.} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I} \end{aligned} \tag{16}$$

To optimize the above minimization in the t th iteration, we use $\Phi(\mathbf{P}_{t-1})$ to estimate the $\Phi(\mathbf{P})$ in Eq. (16), where \mathbf{P}_{t-1} is

the projection matrix acquired in $(t - 1)$ th iteration. Using the eigenvalue decomposition (EVD) technique, we have

$$[\mathbf{U}, \mathbf{S}, \mathbf{U}] = \text{EVD}(\lambda\Phi(\mathbf{P}) + \lambda_3\mathbf{X}\mathbf{L}\mathbf{X}^T) \tag{17}$$

Then, we can update \mathbf{P} as the d eigenvectors in \mathbf{U} associated with the first d smallest eigenvalues in \mathbf{S} , i.e., $\mathbf{P}_t = \mathbf{U}(:, 1 : d)^T$.

4.2 Update of representation matrix R

We then can optimize \mathbf{R} with \mathbf{P} and \mathbf{E} fixed. By ignoring irrelevant terms, the cost function in Eq. (13) reduces to

$$\begin{aligned} \min_{\mathbf{R}} \quad & \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_1 \|\mathbf{R}\|_* + \lambda_2 \|\mathbf{R}\|_1 \\ \text{s.t.} \quad & \mathbf{R}^T\mathbf{1} = \mathbf{1} \end{aligned} \tag{18}$$

To solve the above minimization, we need to introduce two auxiliary variables, \mathbf{J} and \mathbf{S} and reformulate the optimization problem. For brevity, let $\mathbf{A} = \mathbf{P}\mathbf{X}$ and $\mathbf{B} = \mathbf{P}\mathbf{X} - \mathbf{E}$. We can rewrite Eq. (18) as

$$\begin{aligned} \min_{\mathbf{R}} \quad & \lambda \|\mathbf{B} - \mathbf{A}\mathbf{R}\|_F^2 + \lambda_1 \|\mathbf{J}\|_* + \lambda_2 \|\mathbf{S}\|_1 \\ \text{s.t.} \quad & \mathbf{R}^T\mathbf{1} = \mathbf{1}, \mathbf{R} = \mathbf{J}, \mathbf{R} = \mathbf{S}, \end{aligned} \tag{19}$$

We can solve the above problem by using the inexact augmented Lagrangian multiplier (ALM) [42] method. The augmented Lagrangian function of Eq. (19) is

$$\begin{aligned} \min_{\mathbf{R}} \quad & \lambda \|\mathbf{B} - \mathbf{A}\mathbf{R}\|_F^2 + \lambda_1 \|\mathbf{J}\|_* + \lambda_2 \|\mathbf{S}\|_1 \\ & + \text{tr}[\mathbf{A}_1(\mathbf{R}^T\mathbf{1} - \mathbf{1})] + \text{tr}[\mathbf{A}_2(\mathbf{R} - \mathbf{J})] + \text{tr}[\mathbf{A}_3(\mathbf{R} - \mathbf{S})] \\ & + \frac{u_1}{2} \|\mathbf{R}^T\mathbf{1} - \mathbf{1}\|_F^2 + \frac{u_2}{2} \|\mathbf{R} - \mathbf{J}\|_F^2 + \frac{u_3}{2} \|\mathbf{R} - \mathbf{S}\|_F^2 \end{aligned} \tag{20}$$

where \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 are three Lagrangian multipliers and u_1 , u_2 , u_3 are three penalty factors. The optimization of Eq. (20) can be found in Algorithm 1.

4.3 Update of noise matrix E

In order to solve \mathbf{E} , we keep \mathbf{P} and \mathbf{R} fixed. Hence, the objective function in Eq. (13) can be rewritten as

$$\min_{\mathbf{E}} \quad \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_4 \|\mathbf{E}\|_l \tag{21}$$

when the l_1 norm is used for noise, we have

$$\min_{\mathbf{E}} \quad \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_4 \|\mathbf{E}\|_1 \tag{22}$$

This problem can be efficiently solved by shrinkage [48]. Whereas the $l_{2,1}$ norm is used for noise, we have

$$\min_{\mathbf{E}} \lambda \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_4 \|\mathbf{E}\|_{2,1} \quad (23)$$

According to Lemma 1, we can update \mathbf{E} by

$$\mathbf{E} = \operatorname{argmin} \frac{1}{2} \|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \frac{\lambda_4}{2\lambda} \|\mathbf{E}\|_{2,1}.$$

Lemma 1 [49] *Given a matrix $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$, if the optimal solution of*

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{Z} - \mathbf{y}\|_F^2 + \eta \|\mathbf{Z}\|_{2,1}$$

is \mathbf{Z}^* , and then, the i th column of \mathbf{Z}^* is

$$\mathbf{Z}^*(:, i) = \begin{cases} \frac{\|\mathbf{y}_i\|_2 - \eta}{\|\mathbf{y}_i\|_2} \mathbf{y}_i & \text{if } \eta \leq \|\mathbf{y}_i\|_2 \\ 0 & \text{otherwise} \end{cases}$$

4.4 The algorithms

We first describe the algorithmic procedure for solving \mathbf{R} in Algorithm 1. Then, we discuss the whole algorithmic procedure for solving Eq. (13) in Algorithm 2. Once the representation matrix \mathbf{R} is learned, spectral clustering is applied on the affinity matrix $\mathbf{W} = |\mathbf{R}|^T + |\mathbf{R}|$ to obtain the clustering result. The proposed methods are summarized in Algorithm 3.

Algorithm 1 Inexact ALM Algorithm for Equation (20)

Input:

The matrix \mathbf{A} and \mathbf{B} .

Output:

The low-rank and sparse representation matrix \mathbf{J} .

- 1: **Initialize:** $\mathbf{R} = \mathbf{0}, \mathbf{J} = \mathbf{0}, \mathbf{S} = \mathbf{0}, \mathbf{A}_1 = \mathbf{0}, \mathbf{A}_2 = \mathbf{0}, \mathbf{A}_3 = \mathbf{0}, u_1 = 10^{-6}, u_2 = 10^{-6}, u_3 = 10^{-6}, \max_u = 10^6, \epsilon = 10^{-8}, \rho = 1.1$
 - 2: **while** not converged **do**
 - 3: Fix other variables and update \mathbf{R} by

$$\mathbf{R} = (\lambda \mathbf{A}^T \mathbf{A} + u_1 \mathbf{1}\mathbf{1}^T + (u_2 + u_3) \mathbf{I})^{-1} (\lambda \mathbf{A}^T \mathbf{B} + u_1 \mathbf{1}\mathbf{1}^T + u_2 \mathbf{J} + u_3 \mathbf{S} - \mathbf{1}\mathbf{A}_1 - \mathbf{A}_2 - \mathbf{A}_3)$$
 - 4: Fix other variables and update \mathbf{S} by

$$\mathbf{S} = \operatorname{argmin} \frac{\lambda_2}{u_3} \|\mathbf{S}\|_1 + \|\mathbf{R} + \frac{\mathbf{A}_3}{u_3} - \mathbf{S}\|_F^2$$
 - 5: Fix other variables and update \mathbf{J} by

$$\mathbf{J} = \operatorname{argmin} \frac{\lambda_1}{u_2} \|\mathbf{J}\|_* + \|\mathbf{R} + \frac{\mathbf{A}_2}{u_2} - \mathbf{J}\|_F^2$$
 - 6: Update $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ by

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{A}_1 + u_1 (\mathbf{R}^T \mathbf{1} - \mathbf{1}) \\ \mathbf{A}_2 &= \mathbf{A}_2 + u_2 (\mathbf{R} - \mathbf{J}) \\ \mathbf{A}_3 &= \mathbf{A}_3 + u_3 (\mathbf{R} - \mathbf{S}) \end{aligned}$$
 - 7: Update u_1, u_2, u_3 by

$$\begin{aligned} u_1 &= \min(\rho u_1, \max_u) \\ u_2 &= \min(\rho u_2, \max_u) \\ u_3 &= \min(\rho u_3, \max_u) \end{aligned}$$

Check the convergence conditions

$$\|\mathbf{R}^T \mathbf{1} - \mathbf{1}\|_\infty < \epsilon \text{ and } \|\mathbf{R} - \mathbf{J}\|_\infty < \epsilon$$

$$\text{and } \|\mathbf{R} - \mathbf{S}\|_\infty < \epsilon$$
 - 8: **end while**
-

Algorithm 2 Iterative Procedure for Equation (13)

1: **Initialize:**

Projection \mathbf{P} by solving Equation (12)
 noise matrix \mathbf{E} as $\mathbf{E} = \mathbf{0}$

2: **for** $i = 0; i < i_{max}; i++$ **do**

3: Fix \mathbf{P}, \mathbf{E} and optimizing \mathbf{R} by solving Equation (20) using Algorithm 1.

4: Fix \mathbf{R}, \mathbf{E} and optimizing \mathbf{P} by solving Equation (17).

5: Fix \mathbf{P}, \mathbf{R} and optimizing \mathbf{E} by solving Equation (22) or (23).

6: **end for**

7: **return** \mathbf{R}

Algorithm 3 Robust Latent Subspace Clustering by RLSLRSSC

1: Obtain low-rank and sparse representation matrix \mathbf{R} by using Algorithm 2.

2: Normalize the columns of \mathbf{R} as $\mathbf{r}_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|_\infty}$.

3: Construct a similarity graph by $\mathbf{W} = |\mathbf{R}_i| + |\mathbf{R}_i|^T$.

4: Apply spectral clustering to the similarity graph \mathbf{W} .

5: **return** Cluster labels for all data.

4.5 Complexity and convergence analysis

4.5.1 Time complexity

We analyze the time complexity of two main subproblems of RLSLRSSC optimization as follows:

1. The most time-consuming subproblem is computing low-rank and sparse representation. In Algorithm 1, step 5 is the most time-consuming due to singular value decomposition (SVD) with the cost of $O(N)^3$, where N is the number of data samples. The time complexity of computing the matrix inverse in step 3 costs $O(d)^3$, where d is the dimension of latent space. In short, the total time complexity of Algorithm 1 is $i_1 O(N)^3$, where i_1 is the iterative number of this algorithm.
2. For updating the projection matrix \mathbf{P} , the most time-consuming step is EVD. The time complexity of this step is also $O(N)^3$.

Algorithm 1 is a step of Algorithm 2, and as a result, the total time complexity of RLSLRSSC is $i_1 i_2 O(N)^3$, where i_1 is the iterative number of Algorithm 1, and i_2 is the number of iterations of Algorithm 2.

Then, we analyze the time complexities of all the compared methods briefly. For SSC, the time complexity is $iD \times O(N)^3$, where i is the iterative number and D is the size of dimension. For LSR and LRSC, the time complexity is $O(N)^3$. For LRR and BDR, the time complexity is $i O(N)^3$. The time complexity of LSLRSSC and our algorithm is same, which is $i_1 i_2 O(N)^3$.

4.5.2 Convergence analysis

A proof to ensure the convergence of the inexact ALM method can be found in [17], so we assume Algorithm 1 is convergent. Now, we consider the convergence of Algorithm 2.

Proposition 1 *The cost function $J = \lambda\|\mathbf{P}\mathbf{X} - \mathbf{P}\mathbf{X}\mathbf{R} - \mathbf{E}\|_F^2 + \lambda_1\|\mathbf{R}\|_* + \lambda_2\|\mathbf{R}\|_1 + \lambda_3\text{tr}(\mathbf{P}\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}^T) + \lambda_4\|\mathbf{E}\|_l$ converges to a minimum by employing Algorithm 2.*

Proof First, it is easy to prove that $J \geq 0$ for any $\mathbf{P}, \mathbf{R}, \mathbf{E}$.

Then, based on Algorithm 2, assume we have completed t iterations and $\mathbf{P}_t, \mathbf{R}_t, \mathbf{E}_t$ are obtained. In the $(t + 1)$ th iteration, $\mathbf{R}_{t+1} = \text{argmin}_{\mathbf{R}} J(\mathbf{P}_t, \mathbf{R}_t, \mathbf{E}_t)$ after step 3. Therefore, $J(\mathbf{P}_t, \mathbf{R}_{t+1}, \mathbf{E}_t) \leq J(\mathbf{P}_t, \mathbf{R}_t, \mathbf{E}_t)$. Similarly, we also have $\mathbf{P}_{t+1} = \text{argmin}_{\mathbf{P}} J(\mathbf{P}_t, \mathbf{R}_{t+1}, \mathbf{E}_t)$ and $J(\mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{E}_t) \leq J(\mathbf{P}_t, \mathbf{R}_{t+1}, \mathbf{E}_t)$ after step 4. Finally, we have $\mathbf{E}_{t+1} = \text{argmin}_{\mathbf{E}} J(\mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{E}_t)$ and $J(\mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{E}_{t+1}) \leq J(\mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{E}_t)$ after step 5. Hence, $J(\mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{E}_{t+1}) \leq J(\mathbf{P}_t, \mathbf{R}_t, \mathbf{E}_t)$, and we can deduce that RLSLRSSC is convergent.

5 Experimental results

In this section, we carry out several experiments on three face benchmark databases to demonstrate the superior clustering performance of our proposed method RLSLRSSC. We compare our RLSLRSSC with SSC, LSR, LRR, LRSC, BDR, LRSSC, and LSLRSSC. Two measures are employed to evaluate the performance of all approaches, i.e., clustering accuracy (ACC) and normalized mutual information (NMI) [50]. To test the robustness of these methods, we show their results on the databases corrupted by four kinds of noise: Gaussian noise, Laplacian noise, sample-specific corruptions, and Gaussian and Laplacian mixed noise. We describe four kinds of noise experiments in detail in the following paragraphs.

For the first experiment, we add Gaussian noise to the database. The level of noise intensity (LNI) varies from 0 to 5. When the LNI = 0, the original database is used. The corrupted data are constructed by adding Gaussian noise with zero mean and variance $0.01 \times \text{LNI}$.

For the second experiment, we add Laplacian noise to each image. Similar to the first experiment, the LNI varies from 1 to 5. The corrupted data are generated by adding Laplacian noise with the location parameter zero and the scale parameter $0.05 \times \text{LNI}$.

For the third experiment, we add Gaussian noise to some selected samples. The level of corruption percentage (LCP) varies from 1 to 10. We randomly select $\text{LCP} \times 10\%$

percentage of the data samples and add Gaussian noise with the LNI = 2 to them.

For the last experiment, we add two kinds of noise to the images: Gaussian noise and Laplacian noise. The LNI varies from 1 to 5. To construct the contaminated data, Gaussian noise with zeros mean and variance $0.01 \times \text{LNI}$ is added to all images, and then, Laplacian noise with the location parameter zero and the scale parameter $0.05 \times \text{LNI}$ is added.

5.1 Experiment on the ORL database

The ORL database [51] consists of 400 human face images, with 40 distinct subjects containing 10 different images. For each subject, the images were taken under varying lighting conditions, with different facial expressions and facial details at different times. All the images were taken against a dark background with upright, frontal positions. The size of the face images is 32×26 . The images of the ORL database are shown in Fig. 1.

We describe the parameter setting here. The parameters are set to $\lambda = 0.01, \lambda_1 = 0.1, \lambda_2 = 0.01, \lambda_3 = 0.1, \lambda_4 = 0.1$ for the clean data. For the corrupted data, tuning all parameters will cost too much time. Because λ, λ_4 are the coefficients of the error matrix, they indicate the pollution levels. Hence, we fine-tune only λ, λ_4 for the corrupted data. We choose a set of optimal parameters and then repeat five runs and obtain five results and calculate the average result to obtain the final performance.

First, we use the ORL database corrupted by Gaussian noise to demonstrate our method. The performance of related methods is listed in Table 2. When the original images are used, the accuracy of RLSLRSSC-L21 is 86.10%, which is higher than LSLR-SSC (the second place) by 5.40%, which is because RLSLRSSC uses a

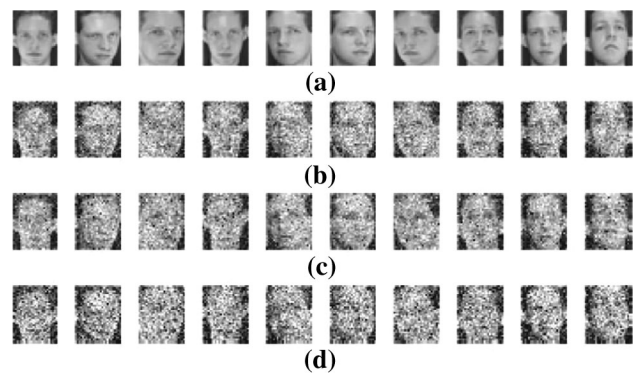


Fig. 1 Samples from the ORL database and contaminated ORL database. **a** The original ORL database. **b** The ORL database corrupted by Gaussian noise. **c** The ORL database corrupted by Laplacian noise. **d** The ORL database corrupted by Gaussian and Laplacian mixed noise. The LNI of the corrupted images is 2

Table 2 Clustering results on the ORL database corrupted by Gaussian noise

Method	Measure	LNI = 0	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	77.65 ± 3.11	79.30 ± 2.04	76.05 ± 2.13	76.30 ± 1.04	72.25 ± 1.84	69.55 ± 2.13
	NMI	88.53 ± 0.77	89.85 ± 0.65	86.76 ± 0.48	86.74 ± 0.79	84.08 ± 0.71	82.29 ± 1.10
LSR1	ACC	74.10 ± 1.40	72.30 ± 1.65	70.20 ± 1.71	70.70 ± 3.46	60.95 ± 2.43	55.80 ± 0.82
	NMI	85.93 ± 0.49	84.49 ± 0.42	83.53 ± 1.07	82.47 ± 1.36	75.62 ± 1.36	71.65 ± 0.71
LSR2	ACC	72.20 ± 2.10	72.10 ± 1.57	69.65 ± 2.39	70.45 ± 2.40	60.85 ± 1.44	56.30 ± 1.24
	NMI	85.81 ± 1.07	84.37 ± 0.55	83.66 ± 1.09	82.56 ± 1.38	75.85 ± 0.50	72.43 ± 1.05
LRR	ACC	73.50 ± 2.78	69.65 ± 2.13	71.10 ± 3.35	69.45 ± 1.78	64.40 ± 1.39	56.35 ± 2.10
	NMI	86.02 ± 1.19	83.36 ± 0.62	83.63 ± 1.27	82.76 ± 0.62	78.28 ± 1.16	72.86 ± 1.15
LRSC	ACC	67.90 ± 1.65	70.40 ± 1.94	63.70 ± 3.16	45.35 ± 2.15	39.55 ± 1.49	38.70 ± 1.19
	NMI	81.51 ± 0.57	83.33 ± 0.81	77.71 ± 1.08	67.74 ± 0.90	63.52 ± 0.74	61.63 ± 0.94
BDR	ACC	72.90 ± 2.21	74.45 ± 1.55	70.15 ± 1.22	64.10 ± 2.27	55.75 ± 1.82	48.70 ± 0.67
	NMI	85.67 ± 1.08	86.28 ± 0.67	82.75 ± 0.54	78.05 ± 0.92	71.84 ± 1.22	66.51 ± 0.39
LRSSC	ACC	79.35 ± 1.91	77.30 ± 0.84	72.60 ± 1.72	64.35 ± 2.45	61.35 ± 1.04	61.50 ± 1.61
	NMI	89.13 ± 0.44	87.86 ± 0.56	84.07 ± 0.86	78.06 ± 1.80	78.20 ± 0.33	79.41 ± 0.83
LSLRSSC	ACC	80.70 ± 2.20	76.45 ± 2.41	71.40 ± 1.04	61.45 ± 1.45	61.65 ± 2.26	63.45 ± 4.30
	NMI	89.04 ± 1.07	87.22 ± 0.79	82.67 ± 0.58	74.65 ± 1.58	77.28 ± 1.36	79.06 ± 1.98
RLSLRSSC-L1	ACC	85.65 ± 0.91	82.90 ± 1.51	82.45 ± 2.16	77.15 ± 2.65	75.10 ± 1.69	74.40 ± 1.62
	NMI	93.45 ± 0.61	91.73 ± 0.50	90.70 ± 0.84	87.63 ± 0.47	85.68 ± 1.12	84.72 ± 0.50
RLSLRSSC-L21	ACC	86.10 ± 1.14	82.30 ± 1.77	81.15 ± 1.81	78.80 ± 1.04	74.00 ± 1.81	73.80 ± 1.78
	NMI	93.61 ± 0.66	91.30 ± 1.02	89.05 ± 0.68	88.03 ± 0.26	85.13 ± 0.73	84.81 ± 0.58

Table 3 Clustering results on the ORL database corrupted by Laplacian noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	79.85 ± 1.44	78.75 ± 1.08	73.45 ± 1.59	60.95 ± 2.19	53.85 ± 1.23
	NMI	89.92 ± 0.88	88.11 ± 0.21	84.52 ± 0.63	76.40 ± 0.99	70.41 ± 1.23
LSR1	ACC	73.85 ± 2.38	72.40 ± 2.28	62.80 ± 1.98	41.75 ± 1.49	30.65 ± 2.24
	NMI	86.06 ± 1.25	85.01 ± 1.61	76.98 ± 0.82	61.84 ± 1.31	51.86 ± 1.18
LSR2	ACC	72.10 ± 3.15	74.15 ± 1.88	63.70 ± 2.35	42.05 ± 2.00	31.50 ± 1.97
	NMI	85.87 ± 1.56	85.90 ± 0.98	77.76 ± 1.10	61.62 ± 1.60	52.20 ± 0.96
LRR	ACC	71.40 ± 3.04	70.75 ± 1.13	65.05 ± 2.38	44.25 ± 1.39	40.85 ± 1.42
	NMI	84.15 ± 1.34	84.32 ± 0.50	79.03 ± 1.78	65.45 ± 0.93	63.23 ± 1.08
LRSC	ACC	70.50 ± 1.48	67.55 ± 2.01	40.30 ± 1.24	37.00 ± 0.85	33.75 ± 0.90
	NMI	83.18 ± 0.70	79.64 ± 1.26	63.85 ± 1.30	60.15 ± 1.10	57.44 ± 0.98
BDR	ACC	75.50 ± 1.98	67.85 ± 1.99	58.50 ± 2.30	34.50 ± 0.94	27.05 ± 0.89
	NMI	86.22 ± 0.97	81.98 ± 0.89	74.23 ± 0.74	55.26 ± 0.82	48.78 ± 0.56
LRSSC	ACC	77.45 ± 1.27	72.50 ± 1.36	61.35 ± 0.80	59.15 ± 1.82	55.55 ± 2.31
	NMI	87.90 ± 0.53	84.46 ± 0.64	78.11 ± 0.95	76.26 ± 0.73	72.83 ± 1.64
LSLRSSC	ACC	78.95 ± 0.80	71.50 ± 1.31	61.05 ± 1.92	61.95 ± 1.49	55.25 ± 1.81
	NMI	88.66 ± 0.55	82.53 ± 0.45	77.85 ± 1.21	77.26 ± 0.88	72.78 ± 0.72
RLSLRSSC-L1	ACC	81.95 ± 2.15	79.95 ± 3.16	76.90 ± 1.07	66.20 ± 2.31	56.55 ± 1.63
	NMI	91.48 ± 0.58	88.74 ± 1.24	87.33 ± 0.72	79.78 ± 1.48	73.16 ± 0.35
RLSLRSSC-L21	ACC	83.65 ± 1.52	80.60 ± 1.49	77.90 ± 2.83	65.60 ± 1.75	58.50 ± 1.58
	NMI	92.73 ± 0.63	89.83 ± 0.25	87.25 ± 1.17	79.29 ± 0.76	74.27 ± 1.00

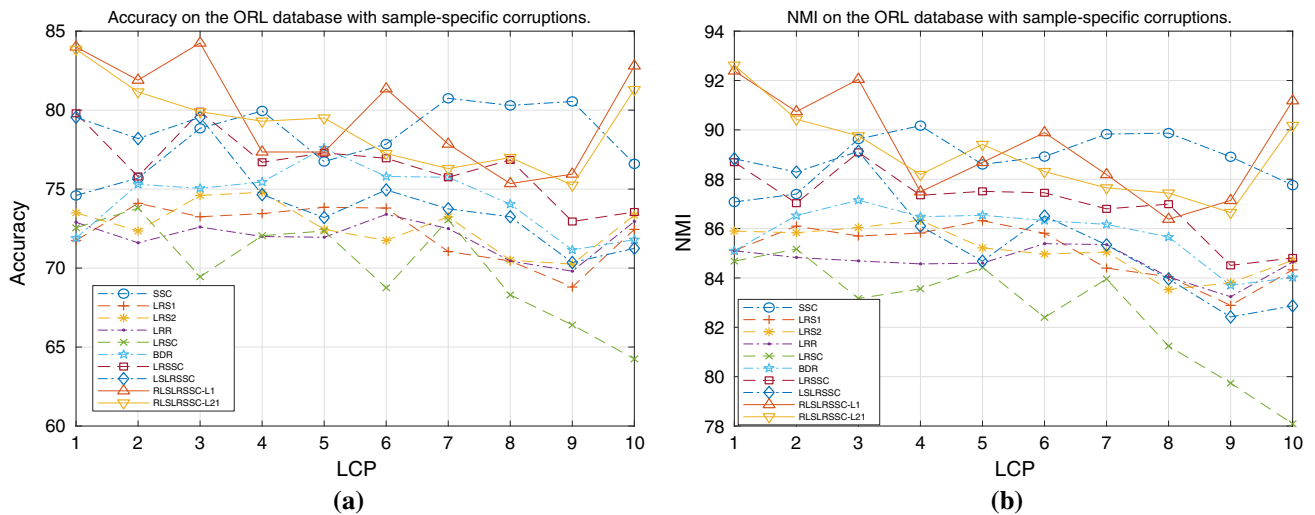


Fig. 2 Clustering performance on the ORL database with sample-specific corruptions. **a** Accuracy; **b** NMI

graph constraint rather than a PCA-like term, so RLSLRSSC can preserve the local manifold structure well. When the LNI increases gradually from 1 to 5, the performance of all the methods decreases. Overall, RLSLRSSC achieves the best result.

We evaluate the performance of related methods on the ORL database corrupted by Laplacian noise. The results are listed in Table 3. As displayed in Table 3, in most cases, RLSLRSSC-L21 can obtain better results. Hence, we deduce that the $l_{2,1}$ norm can capture Laplacian noise well.

We then obtain the clustering performance on the ORL database with sample-specific corruptions. As displayed in Fig. 2, SSC has better results when the LCP is chosen from the set {4, 7, 8, 9}, and our algorithm can obtain competitive results. RLSLRSSC can obtain the best results when the LCP belongs to the set {1, 2, 3, 5, 6, 10}. Overall, RLSLRSSC can deal with sample-specific corruptions well.

As Table 4 shows, RLSLRSSC is robust to Gaussian and Laplacian mixed noise. When the LNI varies from 1 to 4, the performance of RLSLRSSC precedes the competing methods. When the LNI = 5, LSLRSSC has the best performance, RLSLRSSC and LRSSC are competitive, possibly because if we fine-tune only the parameter λ, λ_4 for saving the time of parameter adjustment, a local optimum is obtained.

5.2 Experiments on the UMIST database

The UMIST database [52] consists of 564 images of 20 individuals. We use a cropped version of the UMIST database that is publicly available on the Web page of

University of Sheffield.¹ Each image covers a range of poses from profile to frontal views. We resize the images from 112×92 to 32×32 . The images of UMIST database are shown in Fig. 3.

We set the parameters to $\lambda = 0.003, \lambda_1 = 0.1, \lambda_2 = 0.01, \lambda_3 = 0.1, \lambda_4 = 0.01$ for the clean data. Then, we fine-tune λ, λ_4 for the corrupted data. Also, we choose a set of optimal parameters, repeat five runs, and calculate the average result to get the final performance.

First, we use the UMIST database corrupted by Gaussian noise. Table 5 shows the results of the different methods. When we use the original UMIST database, the accuracy of RLSLRSSC-L21 is 69.01%, which is higher than LRSC (the second place) by 19.65%. When the LNI increases gradually from 1 to 5, RLSLRSSC-L1 achieves the best performance in most cases.

Next, we evaluate the performance of all methods on the UMIST database corrupted by Laplacian noise. The results are displayed in Table 6. When $LNI \leq 3$, RLSLRSSC-L1 obtains the best results, and when $LNI \geq 4$, RLSLRSSC-L21 obtains the best results.

As Fig. 4 shows, the performance of RLSLRSSC on the UMIST database with sample-specific corruptions is better than that of the other methods. When the LCP varies from 1 to 10, the accuracy and NMI of RLSLRSSC are higher than those of the other methods by 8% and 9% at least, respectively. Compared with other algorithms, RLSLRSSC more effectively handles sample-specific corruptions.

Table 7 lists the results of the different methods for the UMIST database corrupted by Gaussian and Laplacian mixed noise. When the UMIST database is corrupted slightly by Gaussian and Laplacian mixed noise, RLSLRSSC performs better than the other methods.

¹ <https://www.sheffield.ac.uk/eee/research/iel/research/face>.

Table 4 Clustering results on the ORL database corrupted by Gaussian and Laplacian mixed noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	80.85 ± 0.55	75.05 ± 1.05	60.15 ± 0.80	52.50 ± 0.73	32.15 ± 1.87
	NMI	89.08 ± 0.25	85.68 ± 0.68	75.26 ± 0.95	69.44 ± 0.41	54.29 ± 1.23
LSR1	ACC	71.90 ± 1.31	67.75 ± 2.97	42.15 ± 1.33	28.80 ± 2.34	20.35 ± 1.02
	NMI	84.52 ± 0.58	79.53 ± 1.36	60.96 ± 0.55	49.81 ± 1.47	43.87 ± 1.01
LSR2	ACC	72.95 ± 1.44	64.45 ± 0.94	42.20 ± 2.41	28.45 ± 1.08	21.75 ± 0.83
	NMI	84.95 ± 0.85	78.38 ± 0.71	60.83 ± 1.68	50.01 ± 0.73	44.66 ± 0.67
LRR	ACC	70.70 ± 1.26	68.20 ± 0.65	46.15 ± 2.10	40.10 ± 2.82	34.95 ± 0.74
	NMI	83.76 ± 0.33	81.30 ± 0.59	66.88 ± 1.50	62.15 ± 1.49	57.57 ± 0.82
LRSC	ACC	72.60 ± 2.33	39.40 ± 1.04	36.85 ± 1.76	33.60 ± 1.43	31.20 ± 0.86
	NMI	83.57 ± 0.75	63.39 ± 0.72	60.38 ± 0.96	57.33 ± 1.23	55.25 ± 0.26
BDR	ACC	73.20 ± 2.43	57.65 ± 1.77	35.40 ± 1.78	23.60 ± 1.71	20.15 ± 1.01
	NMI	85.01 ± 0.99	71.99 ± 1.32	55.44 ± 1.31	46.81 ± 1.48	44.00 ± 0.65
LRSSC	ACC	77.00 ± 2.52	65.15 ± 1.63	58.60 ± 2.45	55.00 ± 2.05	40.45 ± 0.57
	NMI	87.31 ± 1.00	80.93 ± 1.17	76.49 ± 0.69	72.39 ± 1.24	60.98 ± 0.26
LSLRSSC	ACC	75.15 ± 2.01	61.35 ± 2.40	58.50 ± 2.10	55.45 ± 1.87	41.65 ± 0.93
	NMI	85.96 ± 1.07	78.27 ± 1.14	76.54 ± 1.60	71.82 ± 1.01	61.11 ± 1.05
RLSLRSSC-L1	ACC	83.65 ± 1.49	78.90 ± 1.01	65.05 ± 0.99	59.80 ± 2.10	39.40 ± 1.35
	NMI	91.59 ± 0.53	87.42 ± 0.13	79.12 ± 0.51	74.52 ± 1.04	59.47 ± 0.72
RLSLRSSC-L21	ACC	83.05 ± 1.80	79.55 ± 1.02	65.70 ± 0.89	59.40 ± 1.98	38.95 ± 0.69
	NMI	91.57 ± 0.75	87.58 ± 0.29	79.98 ± 0.66	74.18 ± 1.31	58.53 ± 0.45

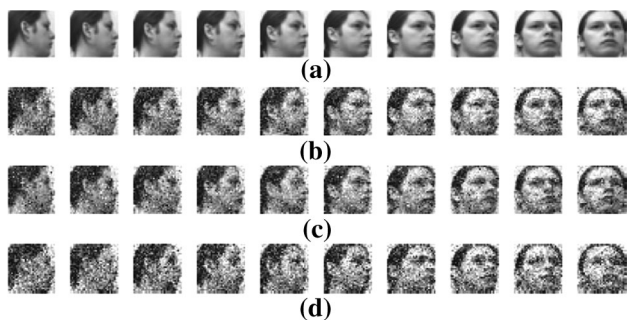


Fig. 3 Samples from the UMIST database and contaminated UMIST database. **a** The original UMIST database. **b** The UMIST database corrupted by Gaussian noise. **c** The UMIST database corrupted by Laplacian noise. **d** The UMIST database corrupted by Gaussian and Laplacian mixed noise. The LNI of the corrupted images is 2

However, when the LNI = 5, the performance of LSLRSSC is the best, which may be for the same reason as for the experiment on the ORL database. Overall, RLSLRSSC is suitable to deal with Gaussian and Laplacian mixed noise.

5.3 Experiment on the CMU face database

The CMU face database consists of 624 black-and-white face images of 20 individuals. The images vary in pose (straight, left, right, up), expression (neutral, happy, sad, angry), eyes (wearing sunglasses or not), and back. The

CMU face images dataset with size 30×32 is obtained from the UCI repository of machine learning databases.² The CMU face database can be found in Fig. 5. We illustrate the 2D visualization of the CMU face by conducting t-sne [53] on the original CMU face database, and the clustering results are shown in Fig. 6).

We set the parameters to $\lambda = 0.001$, $\lambda_1 = 0.1$, $\lambda_2 = 0.01$, $\lambda_3 = 0.01$, $\lambda_4 = 0.001$ for the original CMU face database. Then, we fine-tune λ , λ_4 for the corrupted data. Also, we choose a set of optimal parameters and then repeat five runs and obtain the average result.

First, Gaussian noise is added to the CMU face database. Table 8 shows the results of different methods. RLSLRSSC achieves about 10% better performance than the other methods under all corruption conditions. RLSLRSSC outperforms other algorithms on the CMU face database corrupted by Gaussian noise.

Next, we evaluate the performance of all methods on the CMU face database corrupted by Laplacian noise. The results are displayed in Table 9. Compared with other methods, RLSLRSSC-L1 achieves better results.

We then obtain the clustering results on the CMU face database with sample-specific corruptions displayed in

² <http://archive.ics.uci.edu/ml/datasets/cmu+face+images>.

Table 5 Clustering results on the UMIST database corrupted by Gaussian noise

Method	Measure	LNI = 0	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	45.64 ± 0.38	47.03 ± 1.28	44.84 ± 0.68	47.24 ± 1.56	43.24 ± 0.89	43.30 ± 1.86
	NMI	65.86 ± 0.62	65.56 ± 0.32	64.42 ± 0.88	66.20 ± 0.83	63.22 ± 0.68	62.75 ± 0.94
LSR1	ACC	45.29 ± 1.72	44.42 ± 0.80	44.77 ± 1.47	46.78 ± 1.13	44.56 ± 0.92	43.34 ± 2.87
	NMI	64.00 ± 0.77	63.50 ± 0.51	62.34 ± 1.53	64.41 ± 0.33	62.05 ± 0.50	59.53 ± 1.15
LSR2	ACC	44.28 ± 1.84	44.63 ± 1.10	45.36 ± 1.53	46.82 ± 0.57	42.89 ± 2.11	42.50 ± 1.27
	NMI	64.07 ± 0.82	63.27 ± 0.59	63.34 ± 1.39	64.11 ± 0.46	61.31 ± 0.82	57.83 ± 1.45
LRR	ACC	45.81 ± 0.99	44.00 ± 2.49	46.85 ± 1.52	46.64 ± 1.45	44.42 ± 1.84	42.40 ± 1.67
	NMI	64.52 ± 0.89	63.69 ± 0.92	64.67 ± 0.52	63.70 ± 1.27	61.82 ± 1.03	60.42 ± 0.77
LRSC	ACC	49.36 ± 1.05	46.16 ± 1.70	42.19 ± 1.05	41.18 ± 0.89	41.95 ± 1.16	41.18 ± 1.78
	NMI	66.76 ± 1.19	64.32 ± 0.36	57.90 ± 1.31	57.23 ± 1.17	57.54 ± 0.99	57.97 ± 1.15
BDR	ACC	34.89 ± 1.33	46.71 ± 1.14	45.39 ± 1.32	46.43 ± 2.82	41.98 ± 2.08	34.47 ± 1.98
	NMI	46.54 ± 2.01	62.62 ± 0.46	61.19 ± 0.9	62.18 ± 0.97	55.42 ± 1.02	46.43 ± 1.07
LRSSC	ACC	46.47 ± 1.49	46.47 ± 0.91	48.77 ± 1.25	46.05 ± 1.87	46.43 ± 2.03	44.97 ± 0.61
	NMI	65.64 ± 0.76	65.41 ± 0.65	65.32 ± 0.66	65.87 ± 1.05	64.91 ± 0.59	64.91 ± 1.06
LSLRSSC	ACC	45.43 ± 1.19	44.66 ± 1.21	45.43 ± 1.67	47.97 ± 0.98	47.79 ± 1.64	46.92 ± 2.40
	NMI	64.76 ± 0.44	62.93 ± 0.84	64.69 ± 0.92	60.47 ± 1.03	61.05 ± 1.13	66.05 ± 1.07
RLSLRSSC-L1	ACC	68.00 ± 2.07	62.50 ± 2.16	61.53 ± 3.13	61.98 ± 2.02	57.46 ± 1.37	58.82 ± 2.15
	NMI	83.87 ± 1.83	80.12 ± 1.48	78.98 ± 1.12	78.46 ± 1.16	76.31 ± 0.89	75.97 ± 1.23
RLSLRSSC-L21	ACC	69.01 ± 2.16	62.30 ± 2.06	60.35 ± 1.51	61.84 ± 1.68	57.04 ± 1.61	54.85 ± 1.60
	NMI	84.06 ± 1.27	80.43 ± 1.03	77.65 ± 0.66	79.12 ± 1.13	74.93 ± 1.30	72.53 ± 0.88

Table 6 Clustering results on the UMIST database corrupted by Laplacian noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	48.56 ± 2.37	45.11 ± 1.14	46.16 ± 1.90	43.55 ± 1.68	41.91 ± 1.52
	NMI	67.09 ± 0.66	64.33 ± 0.83	63.91 ± 1.13	63.10 ± 0.61	60.52 ± 0.58
LSR1	ACC	44.90 ± 1.00	44.97 ± 2.08	46.12 ± 0.96	37.95 ± 0.83	29.39 ± 1.36
	NMI	63.69 ± 0.69	63.54 ± 1.10	63.71 ± 0.79	51.86 ± 0.62	35.81 ± 1.05
LSR2	ACC	43.51 ± 0.93	45.32 ± 1.10	45.57 ± 1.10	38.33 ± 2.44	29.36 ± 1.40
	NMI	63.11 ± 0.56	63.01 ± 0.56	63.53 ± 0.72	52.52 ± 1.59	35.71 ± 1.13
LRR	ACC	48.35 ± 2.82	47.27 ± 1.05	44.94 ± 1.41	41.04 ± 1.24	41.60 ± 1.33
	NMI	66.08 ± 1.44	65.64 ± 1.12	61.98 ± 1.12	59.69 ± 0.90	59.36 ± 0.66
LRSC	ACC	48.63 ± 2.75	42.71 ± 2.02	39.86 ± 2.15	41.50 ± 1.19	40.73 ± 1.36
	NMI	65.29 ± 1.22	57.34 ± 0.99	56.55 ± 1.54	58.92 ± 0.36	57.91 ± 0.79
BDR	ACC	43.03 ± 0.72	43.79 ± 1.38	43.72 ± 1.86	29.50 ± 1.63	20.73 ± 1.21
	NMI	60.06 ± 0.71	61.46 ± 1.03	55.99 ± 1.07	36.61 ± 1.39	22.83 ± 1.13
LRSSC	ACC	46.12 ± 2.02	46.23 ± 0.91	45.60 ± 1.44	44.87 ± 0.89	44.00 ± 1.25
	NMI	65.94 ± 1.53	63.76 ± 0.67	65.73 ± 0.65	65.55 ± 0.25	64.50 ± 0.12
LSLRSSC	ACC	46.09 ± 1.62	45.60 ± 2.11	46.71 ± 1.08	44.66 ± 2.26	46.37 ± 2.30
	NMI	66.29 ± 0.34	58.98 ± 1.70	66.09 ± 0.54	65.11 ± 0.63	64.61 ± 1.23
RLSLRSSC-L1	ACC	67.69 ± 1.40	63.97 ± 1.88	61.95 ± 3.81	52.42 ± 1.93	46.26 ± 0.66
	NMI	84.02 ± 0.62	80.14 ± 0.92	78.98 ± 2.26	72.17 ± 1.43	64.12 ± 0.51
RLSLRSSC-L21	ACC	63.93 ± 1.00	60.87 ± 1.73	58.26 ± 1.81	55.06 ± 1.03	46.68 ± 1.00
	NMI	82.46 ± 0.80	79.12 ± 1.15	77.82 ± 1.17	73.15 ± 0.96	64.83 ± 0.95

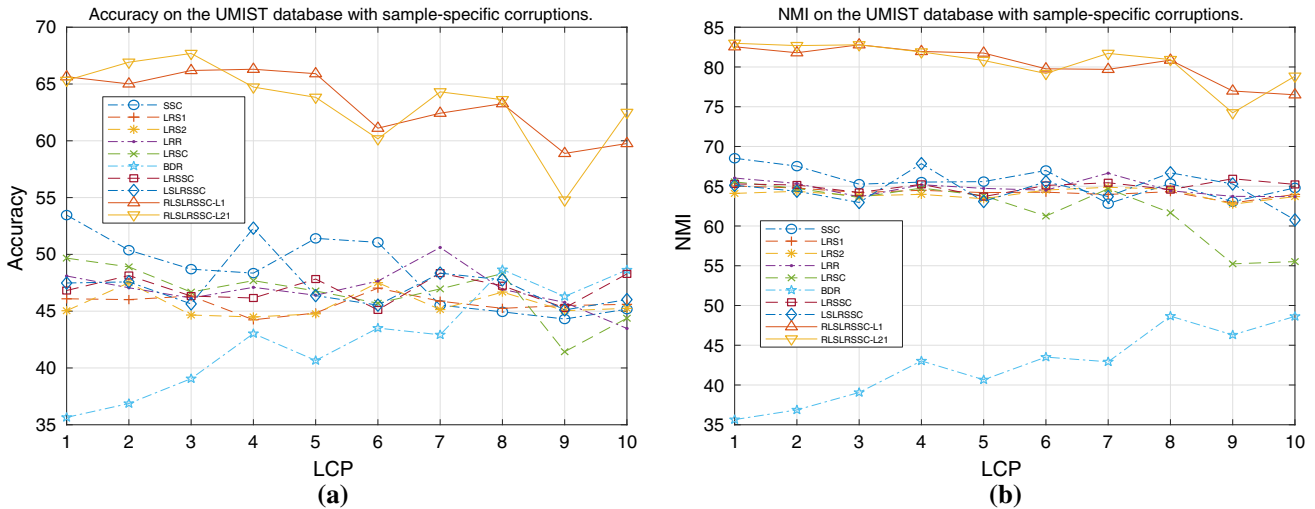


Fig. 4 Clustering performance on the UMIST database with sample-specific corruptions. a Accuracy, b NMI

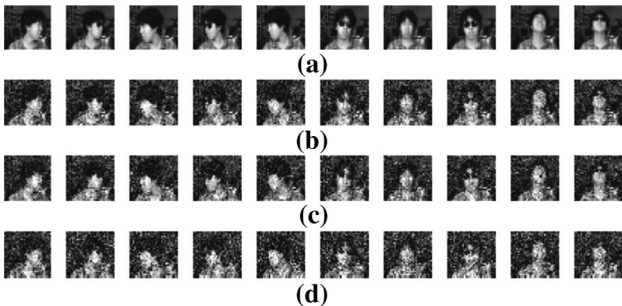


Fig. 5 Samples from the CMU face database and contaminated CMU face database. a The original CMU face database. b The CMU face database corrupted by Gaussian noise. c The CMU face database corrupted by Laplacian noise. d The CMU face database corrupted by Gaussian and Laplacian mixed noise. The LNI of the corrupted images is 2

Fig. 7. We observe that when the LCP varies from 1 to 10, the clustering performances of RLSLRSSC-L1 and RLSLRSSC-L21 are close and stable. We conclude that RLSLRSSC is robust to sample-specific corruptions.

Table 10 lists the results of the different methods on the CMU face database corrupted by Gaussian and Laplacian mixed noise. We can see that RLSLRSSC-L21 yields the best results in most cases.

5.4 Parameter setting

In this subsection, we describe the parameter setting in all experiments. For SSC, the weighting parameter for the l_1 norm is chosen from the set $\{0.1, 1, 10, 20, 50\}$. In LSR, the parameter for $\|Z\|_F$ is chosen from the set $\{0.01, 0.1, 1, 10, 100\}$. In most cases, the parameter is set to 100. In LRR, the parameter for the nuclear norm is selected from

the set $\{0.01, 0.1, 1, 10, 100\}$. In LRSC, the regularization parameter τ is selected from the set $\{0.01, 0.1, 1, 10, 100\}$. The regularization parameter α has the default setting, $\alpha = 0.5 \times \tau$. In BDR, the parameters λ, γ are selected from the set $\{0.001, 0.01, 0.1, 1, 10, 100\}$. In LRSSC, LSLRSSC, and our proposed method RLSLRSSC, the parameter for the nuclear norm λ_1 is selected from the set $\{0.01, 0.1, 1, 10\}$, and the parameter for the l_1 norm λ_2 is $\lambda_2 = 0.1 \times \lambda_1$. In LSLRSSC and RLSLRSSC, the dimension of latent space is fixed at 100.

There are seven parameters in our model, which need to be tuned: $\lambda, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ in Eq. (13), k, t in (10). According to the suggestions of [44], the values of k, t are between 4 and 7, the larger the dataset size, the larger the values of k, t . The values of k, t are 4 because the size of dataset used in experiment is small. In the experiments, we find out that our algorithm can get better performance when $\lambda_2 = 0.1 \times \lambda_1$. Because there are many combinations of remaining four parameters: $\lambda, \lambda_1, \lambda_3, \lambda_4$, we can choose the optimal value of the parameters in two steps. In the first step, we assume that there are few outliers in the clean dataset, so we set the value of λ_4 slightly large, $\lambda_4 = 1$. Then, we fix λ_4 and search for the optimal value of $\lambda, \lambda_1, \lambda_3$. The value of λ is selected from the set $\{0.001, 0.01, 0.1, 1\}$, and the values of λ_1, λ_3 are selected from the set $\{0.01, 0.1, 1, 10\}$. In the second step, we search for the optimal value of λ, λ_4 in $\{0.001, 0.003, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, by fixing other parameters. For the contaminated dataset, we just fine-tune λ, λ_4 for saving time. We found that the optimal value of λ is between 0.001 and 0.01 and the optimal value of λ_4 is between 0.001 and 0.1 for the clean dataset. Hence, for the contaminated dataset, λ is chosen from the set $\{0.001, 0.002, 0.003, 0.004, 0.005,$

Table 7 Clustering results on the UMIST database corrupted by Gaussian and Laplacian mixed noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	45.67 ± 1.82	45.81 ± 1.49	43.62 ± 0.47	41.67 ± 1.55	38.16 ± 1.64
	NMI	64.08 ± 0.96	63.90 ± 0.98	62.06 ± 0.30	58.55 ± 0.96	50.86 ± 1.08
LSR1	ACC	45.81 ± 1.79	44.77 ± 1.83	37.11 ± 1.75	22.50 ± 1.12	19.06 ± 0.87
	NMI	64.45 ± 0.76	61.76 ± 1.15	49.15 ± 1.47	28.18 ± 1.32	20.11 ± 0.46
LSR2	ACC	47.20 ± 2.03	43.30 ± 1.74	36.56 ± 1.34	23.62 ± 1.23	18.02 ± 0.34
	NMI	64.68 ± 0.58	61.20 ± 1.07	49.51 ± 1.31	28.61 ± 0.83	19.25 ± 0.33
LRR	ACC	49.15 ± 0.82	45.01 ± 0.96	41.04 ± 2.35	39.72 ± 1.04	40.28 ± 0.50
	NMI	66.27 ± 1.07	60.54 ± 0.60	58.92 ± 1.70	57.10 ± 0.34	56.46 ± 0.35
LRSC	ACC	46.61 ± 1.71	41.95 ± 1.43	40.42 ± 1.06	40.17 ± 2.15	37.84 ± 1.29
	NMI	62.75 ± 1.13	58.86 ± 0.75	57.57 ± 0.92	55.05 ± 1.34	53.02 ± 0.56
BDR	ACC	48.07 ± 2.60	42.19 ± 0.87	26.75 ± 1.05	19.65 ± 0.97	15.90 ± 0.90
	NMI	63.82 ± 1.64	55.63 ± 1.48	32.32 ± 0.91	20.83 ± 1.18	16.95 ± 0.70
LRSSC	ACC	46.96 ± 0.85	46.02 ± 1.58	43.90 ± 0.57	42.75 ± 0.15	42.47 ± 3.27
	NMI	65.01 ± 0.43	64.69 ± 0.54	64.63 ± 0.64	62.30 ± 0.50	60.25 ± 1.78
LSLRSSC	ACC	41.43 ± 0.81	46.57 ± 1.24	46.61 ± 1.63	42.61 ± 1.45	45.81 ± 1.56
	NMI	59.16 ± 0.79	64.25 ± 0.98	64.54 ± 1.33	62.33 ± 0.41	62.09 ± 0.93
RLSLRSSC-L1	ACC	64.56 ± 1.93	59.79 ± 1.72	53.04 ± 2.88	46.43 ± 0.89	42.44 ± 0.94
	NMI	82.96 ± 0.96	76.25 ± 0.89	72.39 ± 1.38	64.63 ± 0.91	57.34 ± 0.51
RLSLRSSC-L21	ACC	63.41 ± 4.12	53.50 ± 1.34	54.40 ± 1.78	45.43 ± 1.68	42.61 ± 1.25
	NMI	81.76 ± 1.26	72.38 ± 0.96	72.72 ± 0.38	63.09 ± 0.71	56.85 ± 1.02

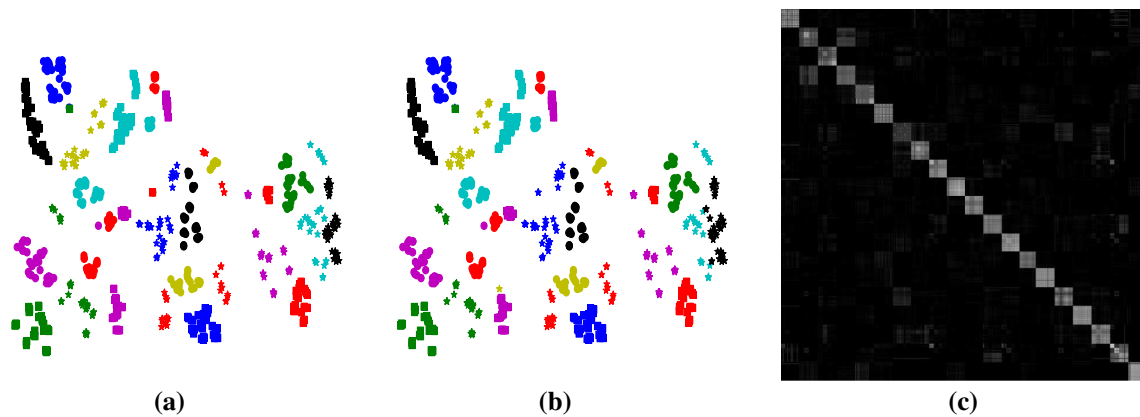


Fig. 6 Clustering result on the CMU face database. **a** Objective result; **b** cluster result; and **c** learned similarity graph

0.01} and λ_4 is chosen from the set {0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1}.

We investigate how sensitive the λ_1, λ_2 is. The value of $\lambda = 0.01, \lambda_3 = 0.1, \lambda_4 = 0.1$ is set for RLSLRSSC, and the effects of the other two parameter are explored. λ_2 has the default value of $\lambda_2 = 0.1 \times \lambda_1$, so we analyze only the effects of λ_1 . We use the original ORL database to conduct our experiment and obtain the average results over five runs. The results are shown in Fig. 8. The performances of RLSLRSSC-L1 and RLSLRSSC-L21 are very close. When λ_1 increases from 0.001 to 0.1, the performance of RLSLRSSC improves remarkably. When λ_1 increases from

0.1 to 1, the performance decreases sharply. Hence, an optimal value can be discovered easily when $\lambda_1 = 0.1$.

Then, to research the parameter sensitivity of λ_3 , we set the value of $\lambda = 0.01, \lambda_1 = 0.1, \lambda_2 = 0.01, \lambda_4 = 0.1$ for RLSLRSSC and study the effects of λ_3 . We also use the original ORL database to conduct the experiment and get the average results over five runs. As shown in Fig. 9, the performance of RLSLRSSC is stable when $\lambda_3 \geq 0.05$. We notice that when $\lambda_3 = 0$, the performance decreases remarkably, which shows the importance of the graph constraint term.

Table 8 Clustering results on the CMU face database corrupted by Gaussian noise

Method	Measure	LNI = 0	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	43.53 ± 2.29	65.67 ± 2.43	76.51 ± 2.15	77.53 ± 1.05	74.52 ± 2.82	75.35 ± 0.87
	NMI	59.11 ± 2.48	77.36 ± 1.22	83.50 ± 0.95	83.79 ± 0.52	82.69 ± 1.49	81.43 ± 0.91
LSR1	ACC	87.60 ± 2.89	80.96 ± 0.44	79.84 ± 4.52	74.78 ± 0.62	75.45 ± 2.15	72.69 ± 3.69
	NMI	91.05 ± 0.88	86.33 ± 0.28	85.59 ± 2.85	81.89 ± 0.41	82.21 ± 1.19	80.51 ± 2.16
LSR2	ACC	86.64 ± 1.90	81.41 ± 4.07	80.74 ± 4.63	76.31 ± 1.75	74.39 ± 1.75	75.00 ± 2.19
	NMI	91.28 ± 1.25	86.90 ± 2.85	85.73 ± 2.13	82.60 ± 0.89	81.98 ± 1.82	81.47 ± 1.16
LRR	ACC	78.94 ± 3.28	86.25 ± 2.15	86.64 ± 2.88	81.28 ± 2.33	78.08 ± 1.20	77.82 ± 1.96
	NMI	88.13 ± 2.45	90.07 ± 0.75	90.20 ± 1.03	85.63 ± 1.67	82.93 ± 0.53	83.12 ± 0.82
LRSC	ACC	82.40 ± 3.01	76.57 ± 3.42	75.26 ± 2.37	63.17 ± 1.71	50.61 ± 1.92	50.48 ± 1.13
	NMI	89.95 ± 1.46	85.70 ± 1.84	82.97 ± 1.76	68.80 ± 0.81	62.50 ± 1.01	62.23 ± 0.41
BDR	ACC	47.40 ± 1.75	76.63 ± 3.67	81.60 ± 1.46	76.12 ± 3.79	69.30 ± 3.21	67.82 ± 2.34
	NMI	58.01 ± 0.67	82.52 ± 1.90	86.26 ± 1.17	82.29 ± 1.27	77.48 ± 2.12	73.44 ± 1.58
LRSSC	ACC	80.32 ± 0.31	78.72 ± 1.49	81.31 ± 3.74	75.77 ± 1.74	67.76 ± 2.48	69.42 ± 0.52
	NMI	85.23 ± 0.20	84.54 ± 1.14	86.20 ± 1.54	82.45 ± 0.77	74.44 ± 1.46	77.90 ± 0.29
LSLRSSC	ACC	79.84 ± 0.63	76.25 ± 3.35	75.10 ± 2.97	76.22 ± 0.77	72.56 ± 1.77	70.06 ± 1.37
	NMI	84.52 ± 0.55	82.44 ± 1.89	82.04 ± 1.82	81.38 ± 0.24	81.38 ± 0.82	78.86 ± 0.52
RLSLRSSC-L1	ACC	95.87 ± 1.42	92.50 ± 2.81	95.87 ± 1.20	84.87 ± 2.35	83.72 ± 0.85	86.15 ± 0.62
	NMI	96.83 ± 0.74	94.00 ± 1.90	96.66 ± 0.86	90.62 ± 1.23	89.63 ± 0.33	89.86 ± 0.47
RLSLRSSC-L21	ACC	95.03 ± 0.00	94.71 ± 0.00	91.12 ± 3.53	86.41 ± 1.26	84.55 ± 1.45	83.65 ± 0.73
	NMI	96.41 ± 0.00	95.79 ± 0.18	93.81 ± 1.76	89.65 ± 0.45	88.41 ± 0.84	89.74 ± 0.56

Table 9 Clustering results on the CMU face database corrupted by Laplacian noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	61.57 ± 1.54	75.22 ± 0.37	75.00 ± 2.91	74.01 ± 2.46	70.42 ± 1.75
	NMI	74.05 ± 1.24	82.39 ± 0.33	82.98 ± 1.70	81.86 ± 1.17	76.21 ± 0.94
LSR1	ACC	86.64 ± 2.93	80.26 ± 3.58	77.24 ± 1.41	75.96 ± 3.27	59.97 ± 1.58
	NMI	90.30 ± 1.68	86.41 ± 2.15	83.57 ± 0.88	80.81 ± 1.64	61.88 ± 0.34
LSR2	ACC	84.49 ± 1.55	80.16 ± 3.43	77.34 ± 1.71	76.25 ± 3.60	58.17 ± 1.29
	NMI	88.83 ± 0.8	85.66 ± 2.53	83.99 ± 0.91	80.76 ± 2.50	59.75 ± 1.48
LRR	ACC	84.01 ± 2.27	87.18 ± 3.18	81.80 ± 1.70	70.93 ± 1.96	53.97 ± 2.69
	NMI	90.04 ± 1.09	90.68 ± 1.75	86.24 ± 1.00	75.91 ± 0.83	65.94 ± 1.29
LRSC	ACC	82.82 ± 1.39	75.32 ± 2.36	44.97 ± 1.07	51.76 ± 0.96	45.22 ± 1.51
	NMI	89.40 ± 1.99	83.65 ± 1.59	58.79 ± 1.22	61.53 ± 0.78	58.00 ± 0.76
BDR	ACC	74.01 ± 2.97	83.43 ± 1.92	72.66 ± 1.35	59.04 ± 2.13	32.40 ± 1.90
	NMI	79.82 ± 1.62	86.63 ± 1.20	76.99 ± 0.82	63.00 ± 1.52	34.74 ± 0.89
LRSSC	ACC	76.86 ± 2.54	82.98 ± 1.15	73.49 ± 2.50	71.92 ± 2.77	69.94 ± 1.93
	NMI	84.82 ± 2.11	88.06 ± 0.49	76.65 ± 0.78	78.93 ± 1.73	77.60 ± 0.98
LSLRSSC	ACC	75.58 ± 1.62	77.18 ± 2.99	78.56 ± 2.36	70.16 ± 0.70	72.72 ± 2.25
	NMI	83.17 ± 0.67	83.46 ± 0.56	83.09 ± 1.36	77.68 ± 0.77	77.97 ± 1.28
RLSLRSSC-L1	ACC	96.60 ± 0.07	93.30 ± 2.55	86.03 ± 1.86	85.03 ± 2.34	82.02 ± 1.28
	NMI	97.08 ± 0.11	95.27 ± 0.98	90.08 ± 0.75	88.72 ± 1.36	84.55 ± 1.08
RLSLRSSC-L21	ACC	95.35 ± 0.93	91.83 ± 0.49	85.51 ± 0.61	84.23 ± 1.44	80.13 ± 3.49
	NMI	96.03 ± 0.59	93.61 ± 0.24	89.86 ± 0.68	87.74 ± 0.61	84.67 ± 1.58

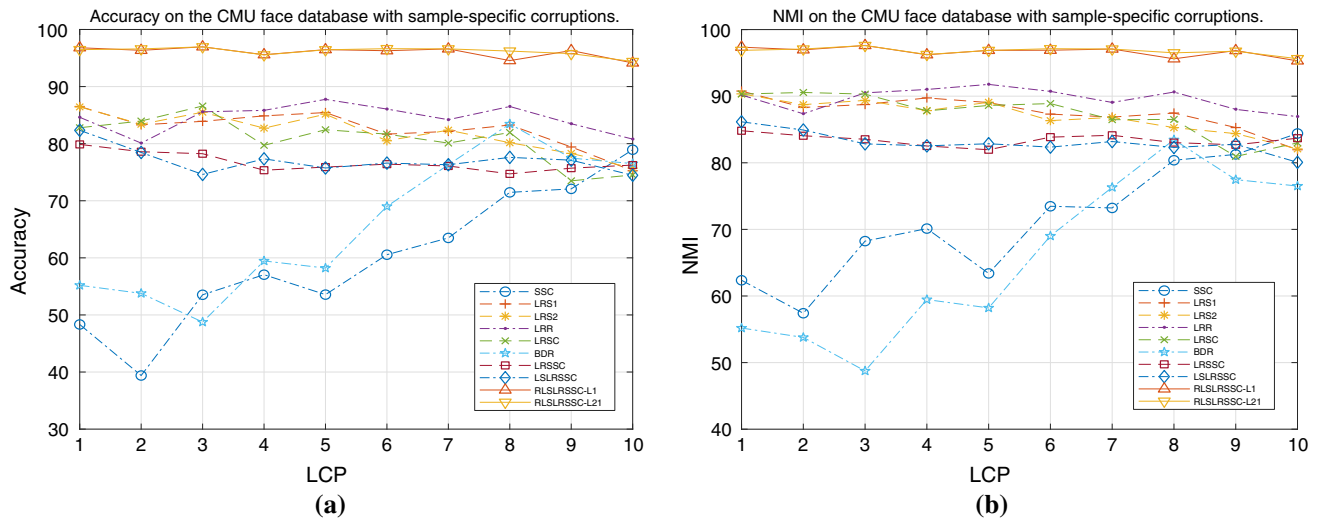


Fig. 7 Clustering performance on the CMU face database with sample-specific corruptions. **a** Accuracy and **b** NMI

Table 10 Clustering results on the CMU face database corrupted by Gaussian and Laplacian mixed noise

Method	Measure	LNI = 1	LNI = 2	LNI = 3	LNI = 4	LNI = 5
SSC	ACC	72.82 ± 3.02	74.81 ± 1.96	72.85 ± 0.82	68.21 ± 1.80	62.47 ± 1.46
	NMI	82.30 ± 1.47	82.78 ± 1.45	79.65 ± 0.42	72.46 ± 1.07	69.11 ± 0.60
LSR1	ACC	78.85 ± 4.18	75.55 ± 1.43	72.53 ± 2.90	45.80 ± 0.50	24.62 ± 0.98
	NMI	85.03 ± 2.57	81.47 ± 0.87	76.30 ± 2.06	48.62 ± 0.64	25.55 ± 0.61
LSR2	ACC	82.15 ± 3.05	74.87 ± 0.24	74.65 ± 1.07	46.73 ± 0.77	20.58 ± 1.16
	NMI	86.89 ± 1.49	81.26 ± 0.21	77.96 ± 0.65	49.08 ± 0.70	23.33 ± 1.36
LRR	ACC	86.92 ± 4.24	54.87 ± 1.71	50.93 ± 1.88	49.26 ± 1.96	52.47 ± 1.02
	NMI	90.19 ± 1.91	67.89 ± 1.44	62.86 ± 1.32	61.95 ± 1.97	62.99 ± 0.53
LRSC	ACC	80.86 ± 3.75	48.05 ± 2.17	43.46 ± 1.83	42.50 ± 1.72	43.30 ± 0.80
	NMI	87.48 ± 2.43	60.41 ± 1.15	56.45 ± 1.24	54.85 ± 1.22	53.43 ± 0.61
BDR	ACC	82.85 ± 0.89	69.33 ± 3.58	50.80 ± 1.26	26.03 ± 2.25	16.92 ± 1.12
	NMI	87.87 ± 1.72	75.67 ± 1.50	54.17 ± 0.74	27.88 ± 1.66	16.98 ± 1.24
LRSSC	ACC	79.46 ± 3.46	72.66 ± 1.77	66.31 ± 1.74	67.50 ± 1.08	62.28 ± 1.89
	NMI	85.93 ± 0.84	77.80 ± 0.95	74.77 ± 0.81	76.08 ± 0.57	70.11 ± 1.79
LSLRSSC	ACC	78.43 ± 2.99	76.57 ± 2.07	68.43 ± 1.28	69.30 ± 2.16	66.28 ± 0.92
	NMI	83.53 ± 1.90	82.09 ± 0.68	75.59 ± 0.71	76.48 ± 0.82	73.58 ± 0.56
RLSLRSSC-L1	ACC	95.55 ± 0.90	86.96 ± 0.64	80.00 ± 2.79	71.70 ± 1.05	68.05 ± 0.48
	NMI	96.02 ± 0.71	91.32 ± 0.54	84.04 ± 1.14	79.08 ± 0.66	72.33 ± 0.52
RLSLRSSC-L21	ACC	96.44 ± 0.13	88.40 ± 2.09	81.89 ± 2.50	74.17 ± 1.56	68.75 ± 1.32
	NMI	96.85 ± 0.20	90.73 ± 1.39	86.45 ± 0.79	77.01 ± 1.46	73.30 ± 0.31

Finally, we investigate the clustering performance on the ORL database changes by varying the regularization parameter λ, λ_4 for RLSLRSSC-L1 and RLSLRSSC-L21. We set the value of $\lambda_1 = 0.1, \lambda_2 = 0.01, \lambda_3 = 0.1$ and then explore the effects of the other two parameters. We use the ORL database corrupted by Gaussian and Laplacian mixed noise with the value of the LNI = 3. For each set of parameters, we average the results over five runs. Figure 10 illustrates the final results. The larger the error, the smaller

the weight, i.e., if the value of λ or λ_4 decreases, the error increases. The performance of RLSLRSSC-L1 is improved to a certain extent when the value of λ or λ_4 decreases, so the terms associated with λ and λ_4 are proved to work. RLSLRSSC-L21 is not as stable as RLSLRSSC-L1. When $\lambda = 0.001, \lambda_4$ decreases from 0.005 to 0.001, the performance of RLSLRSSC-L21 declines substantially. In brief, RLSLRSSC-L1 performs stably with different parameters and addresses Gaussian and Laplacian mixed noise well.

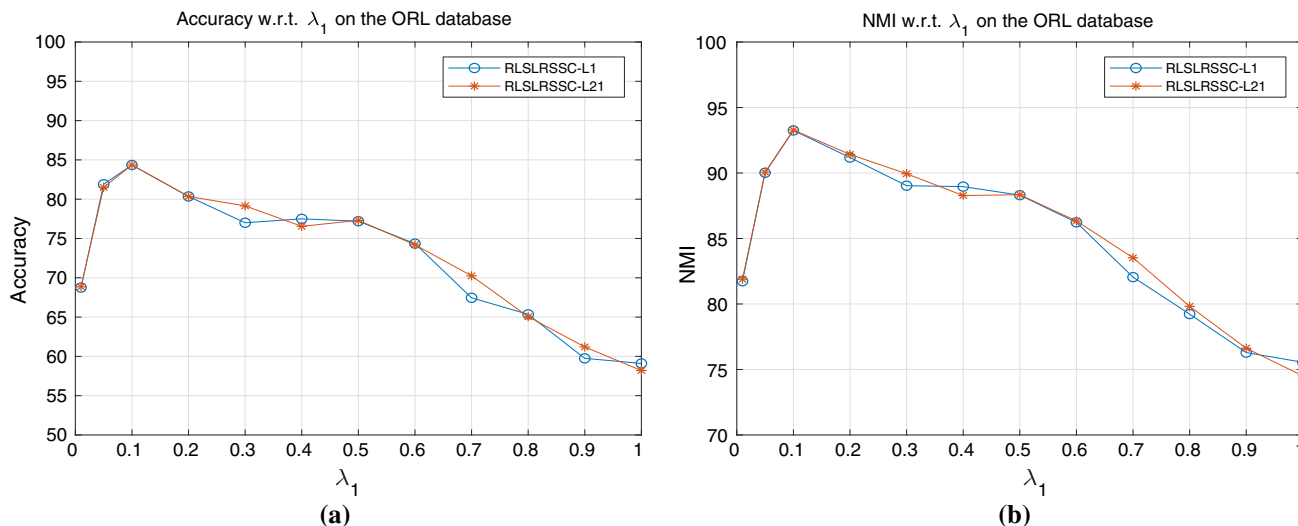


Fig. 8 Clustering performance with different values of λ_1 on the ORL database. **a** Accuracy and **b** NMI

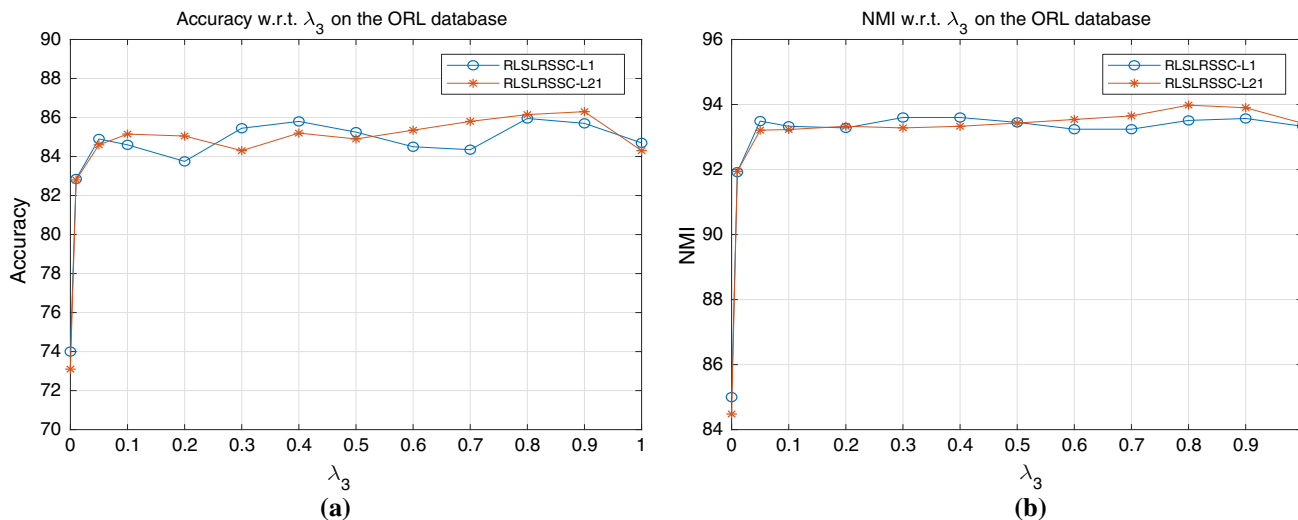


Fig. 9 Clustering performance with different values of λ_3 on the ORL database. **a** Accuracy and **b** NMI

6 Conclusion and future works

This work proposes a novel subspace clustering method, RLSLRSSC, to deal with the corrupted face database clustering problem. The F-norm and the l_1 norm or the $l_{2,1}$ are used successfully to capture the noise and outliers. RLSLRSSC simultaneously learns a robust projection and a low-rank and sparse representation in the low-dimensional subspace. By incorporating a graph constraint, the local manifold structure can be preserved, and the discriminative ability of the learned projection can be promoted. We also present adequate experiments to

demonstrate that RLSLRSSC is superior to other state-of-the-art methods.

There is, however, still an unsolved problem. To obtain a discriminative latent space, we introduced a graph constraint in our model. The high-dimensional data can be represented by the low-dimensional embedding in the learned latent space. In fact, we get the low-dimensional embedding in the manner of linear transformation. For the data with nonlinear structure, our method may not work well. In the recent years, deep learning has yielded fruitful results. The data can be nonlinearly mapped into a latent space by using deep learning. In the future, we plan to combine our method with deep learning.

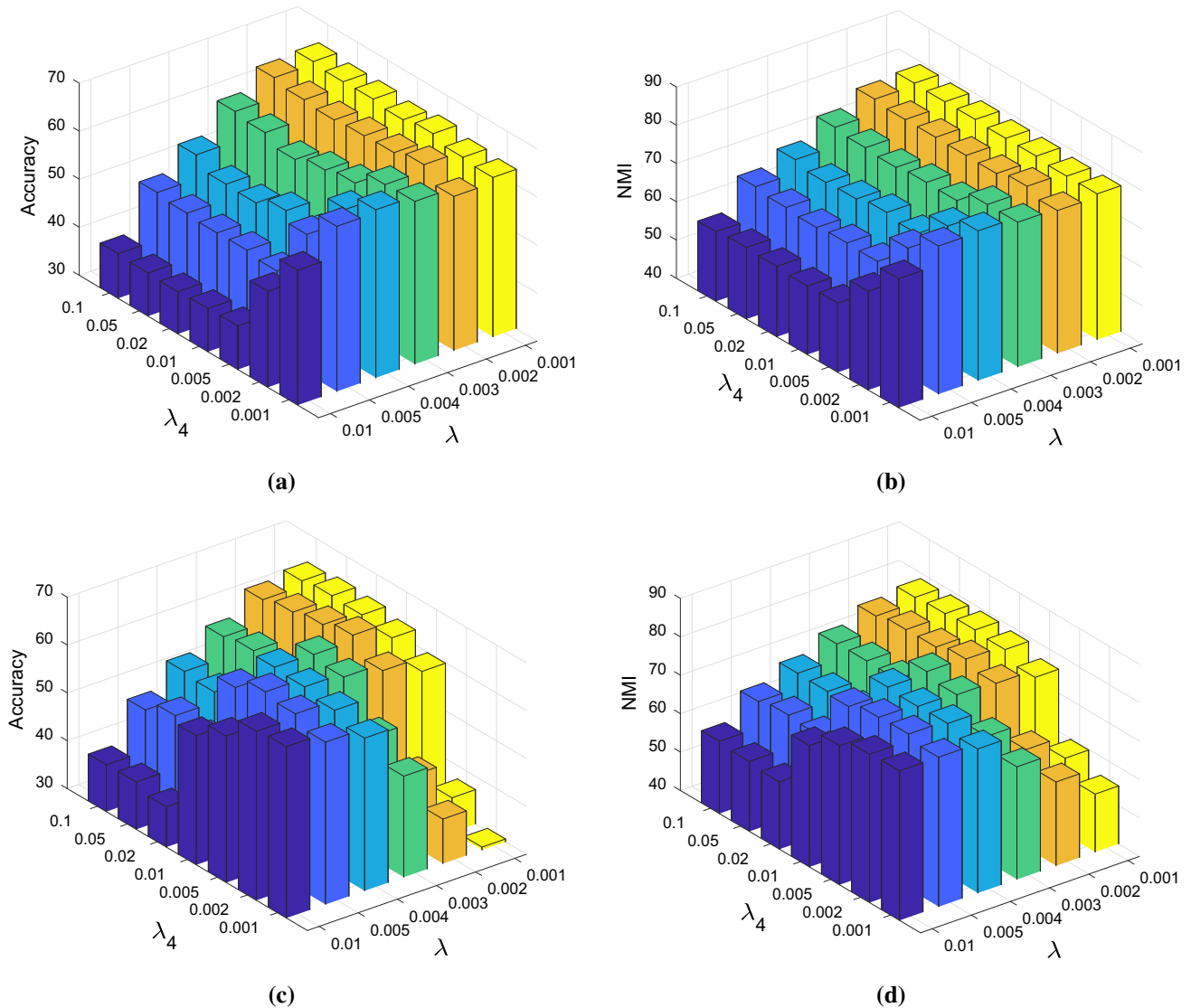


Fig. 10 Clustering performance with different values of λ , λ_4 on the ORL database corrupted by Gaussian and Laplacian mixed noise with the LNI = 3. **a** Accuracy for RLSLRSSC-L1. **b** NMI for RLSLRSSC-L1. **c** Accuracy for RLSLRSSC-L21. **d** NMI for RLSLRSSC-L21

Acknowledgements This work is supported by the National Natural Science Foundation of China (61402181, 61502174), the Natural Science Foundation of Guangdong Province (2015A030313215, 2017A030313358, 2017A030313355), the Science and Technology Planning Project of Guangdong Province (2016A040403046), the Guangzhou Science and Technology Planning Project (201704030051).

Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Basri R, Jacobs D (2001) Lambertian reflectance and linear subspaces. In: ICCV
2. Agarwal P, Mustafa N (2004) k-means projective clustering. In: ACM symposium on principles of database systems, pp 155–165
3. Zhang T, Szlam A, Lerman G (2009) Median k-flats for hybrid linear modeling with many outliers. In: ICCV
4. Zhang T, Szlam A, Wang Y, Lerman G (2012) Hybrid linear modeling via local best-fit flats. *Int J Comput Vis* 100(3):217–240
5. Huang K, Ma Y, Vidal R (2004) Minimum effective dimension for mixtures of subspaces: a robust GPCA algorithm and its applications. In: CVPR
6. Ma Y, Yang AY, Derksen H, Fossum R (2008) Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Rev* 50(3):413–458
7. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). *IEEE Trans Pattern Anal Mach Intell* 27(12):1–15

8. Archambeau C, Delannay N, Verleysen M (2008) Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* 71(7–9):1274–1282
9. Gruber A, Weiss Y (2004) Multibody factorization with uncertainty and missing data using the EM algorithm. In: *CVPR*
10. Ma Y, Derksen H, Hong W, Wright J (2007) Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Trans Pattern Anal Mach Intell* 29(9):1546–1562
11. Yang AY, Rao SR, Ma Y (2006) Robust statistical estimation and segmentation of multiple subspaces. In: *CVPR*
12. Chen G, Lerman G (2009) Spectral curvature clustering (SCC). *Int J Comput Vision* 81(3):317–330
13. Elhamifar E, Vidal R (2009) Sparse subspace clustering. In: *CVPR*
14. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell* 35(11):2765–2781
15. Favaro P, Vidal R, Ravichandran A (2011) A closed form solution to robust subspace estimation and clustering. In: *CVPR*
16. Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: *ICML*
17. Liu G, Lin Z, Yan S, Sun J, Ma Y (2013) Robust recovery of subspace structures by low-rank representation. *IEEE Trans Pattern Anal Mach Intell* 35(1):171–184
18. Lu C, Lin Z, Yan S (2013) Correlation adaptive subspace segmentation by trace lasso. In: *ICCV*
19. Lu C, Min H, Zhao ZQ, Zhu L, Huang DS, Yan S (2012) Robust and efficient subspace segmentation via least squares regression. In: *ECCV*
20. Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). *Pattern Recognit Lett* 43(1):47–61
21. Wang YX, Xu H, Leng C (2013) Provable subspace clustering: when LRR meets SSC. In: *NIPS*
22. Chen J, Yang J (2014) Robust subspace segmentation via low-rank representation. *IEEE Trans Cybern* 44(8):1432–1445
23. Donoho DL, Elad M, Temlyakov VN (2006) Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans Inf Theory* 52(1):6–18
24. Yang Y, Feng J, Jojic N, Yang J, Huang TS (2016) L0-sparse subspace clustering. In: *ECCV*
25. Li J, Kong Y, Fu Y (2017) Sparse subspace clustering by learning approximation L0 codes. In: *AAAI*
26. Yin M, Xie S, Wu Z, Zhang Y, Gao J (2018) Subspace clustering via learning an adaptive low-rank graph. *IEEE Trans Image Process* 27(8):3716–3728
27. Lu C, Feng J, Lin Z, Mei T, Yan S (2018) Subspace clustering by block diagonal representation. *IEEE Trans Pattern Anal Mach Intell* 41(2):487–501
28. Jolliffe IT (2002) *Principal component analysis*, 2nd edn. Springer, New York
29. Bingham E, Mannila H (2001) Random projection in dimensionality reduction: applications to image and text data. In: *ACM SIGKDD international conference on knowledge discovery and data mining*, pp 245–250
30. Patel VM, Nguyen HV, Vidal R (2013) Latent space sparse subspace clustering. In: *ICCV*
31. Patel VM, Nguyen HV, Vidal R (2015) Latent space sparse and low-rank subspace clustering. *IEEE J Sel Topics Signal Process* 9(4):691–701
32. Wei L, Wu A, Yin J (2015) Latent space robust subspace segmentation based on low-rank and locality constraints. *Expert Syst Appl* 42(19):6598–6608
33. He X, Niyogi P (2004) Locality preserving projections. In: *NIPS*
34. He X, Yan S, Hu Y, Niyogi P, Zhang H (2005) Face recognition using laplacianfaces. *IEEE Trans Pattern Anal Mach Intell* 27(3):328–340
35. Tang K, Su Z, Jiang W, Zhang J, Sun X, Luo X (2018) Robust subspace learning-based low-rank representation for manifold. *Neural Comput Appl* 29:329
36. Jalali A, Sujay S, Ruan C, Ravikumar PK (2010) A dirty model for multi-task learning. In: *NIPS*
37. Wang D, Liu Q, Xia Y, Dong P, Luo J, Huang Q, Feng DD (2013) Dictionary learning based impulse noise removal via L1–L1 minimization. *Signal Process* 93(9):2696–2708
38. Chen Z, Wu Y (2013) Robust dictionary learning by error source decomposition. In: *ICCV*
39. Zhang H, Wang S, Zhao M, Xu X, Ye Y (2018) Locality reconstruction models for book representation. *IEEE Trans Knowl Data Eng* 30(10):1873–1886
40. Zhang H, Wang S, Xu X, Chow TWS, Wu QMJ (2018) Tree2Vector: learning a vectorial representation for tree-structured data. *IEEE Trans Neural Netw Learn Syst* 29(11):5304–5318
41. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
42. Lin Z, Chen M, Wu L, Ma Y (2009) The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC technical report UILU-ENG-09-2215
43. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: analysis and an algorithm. In: *NIPS*
44. Zelnik-Manor L, Perona P (2005) Self-tuning spectral clustering. In: *NIPS*
45. Cai D, He X (2005) Orthogonal locality preserving indexing. In: *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pp 3–10
46. Kokiopoulou E, Saad Y (2007) Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. *IEEE Trans Pattern Anal Mach Intell* 29(12):2143–2156
47. Cai D, He X, Han J, Zhang H (2006) Orthogonal laplacianfaces for face recognition. *IEEE Trans Image Process* 15(11):3608–3614
48. Hale E, Yin W, Zhang Y (2008) Fixed-point continuation for l1-minimization: methodology and convergence. *SIAM J Optim* 19(3):1107–1130
49. Yang J, Yin W, Zhang Y, Wang Y (2009) A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM J Imaging Sci* 2(2):569–592
50. Cai D, He X, Han J (2007) Spectral regression: a unified approach for sparse subspace learning. In: *ICDM*
51. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: *IEEE workshop on applications of computer vision*
52. Phillips J, Bruce V, Soulie FF (1999) *In face recognition: from theory to applications*. Springer, Berlin
53. Maaten LVD, Hinton G (2008) Visualizing data using T-SNE. *J Mach Learn Res* 9:2579–2605

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