Robust Kernel Low-Rank Representation

Shijie Xiao, *Student Member, IEEE*, Mingkui Tan, *Member, IEEE*, Dong Xu, *Senior Member, IEEE*, and Zhao Yang Dong, *Senior Member, IEEE*

***Abstract*— Recently, low-rank representation (LRR) has shown promising performance in many real-world applications such as face clustering. However, LRR may not achieve satisfactory results when dealing with the data from nonlinear subspaces, since it is originally designed to handle the data from linear subspaces in the input space. Meanwhile, the kernel-based methods deal with the nonlinear data by mapping it from the original input space to a new feature space through a kernel-induced mapping. To effectively cope with the nonlinear data, we first propose the kernelized version of LRR in the clean data case. We also present a closed-form solution for the resultant optimization problem. Moreover, to handle corrupted data, we propose the robust kernel LRR (RKLRR) approach, and develop an efficient optimization algorithm to solve it based on the alternating direction method. In particular, we show that both the subproblems in our optimization algorithm can be efficiently and exactly solved, and it is guaranteed to obtain a globally optimal solution. Besides, our proposed algorithm can also solve the original LRR problem, which is a special case of our RKLRR when using the linear kernel. In addition, based on our new optimization technique, the kernelization of some variants of LRR can be similarly achieved. Comprehensive experiments on synthetic data sets and real-world data sets clearly demonstrate the efficiency of our algorithm, as well as the effectiveness of RKLRR and the kernelization of two variants of LRR.**

***Index Terms*— Low-rank representation (LRR), kernel methods.**

I. Introduction

CLUSTERING is one of the essential tasks in machine learning and data mining. In particular, given data sampled from a union of subspaces, subspace clustering [2]–[5] is to partition data into several clusters, so that each cluster corresponds to one subspace. The existing subspace clustering methods can be roughly classified into the following four categories: 1) iterative approaches [6]; 2) statistical approaches [7]; 3) algebraic approaches [8]; and 4) spectral clustering-based approaches [1], [9]–[11]. In particular, the spectral clustering-based approaches first seek a desired affinity matrix containing pairwise affinities between all data points, and then apply spectral clustering (e.g., NCut [12]) on it to obtain the clustering result.

Among these spectral clustering-based approaches, both sparse subspace clustering (SSC) [10] and low-rank representation (LRR) [1] seek a desired affinity matrix by learning a data representation matrix. In particular, they both use the self-expressiveness property of data (i.e., each data point in a union of subspaces can be efficiently represented as a linear combination of other points). In particular, SSC seeks a sparse representation of data [13]. However, as mentioned in [1], SSC may not accurately capture the global structures of data, especially when data are grossly corrupted. To capture the global structures of data, LRR [1] encourages the data representation matrix to be low rank. LRR has shown promising performance in different applications [14], and many variants [15] of LRR have been proposed.

LRR [1] is originally proposed to deal with the data from multiple linear subspaces on the input space [16], and it may not achieve satisfactory results when dealing with the data from nonlinear subspaces [Fig. 1(a) and (b)]. Using kernel-induced mapping [17] to map the data from the original input space to a feature space [16], the mapped data in the feature space may reside in multiple linear subspaces. In this paper, we propose a new kernelized version of LRR to deal with such data. In particular, we provide a closed-form solution regarding the kernelized version of LRR for handling clean data. Moreover, to facilitate the kernelization of LRR for dealing with corrupted data, we first rewrite the optimization problem of LRR in a new form, where the data matrix $\mathbf{X}$ appears only in the form of inner product (i.e., $\mathbf{X}'\mathbf{X}$). Based on this new form, we propose the robust kernel LRR (RKLRR) method, which essentially performs LRR in the new feature space, with the $\ell_{2,1}$ norm regularization [1] on the representation error in order to make the RKLRR robust to outliers.

To solve the optimization problem of RKLRR with a convex but nonsmooth objective function, we adopt the alternating direction method (ADM) with two blocks of variables [18]–[20], which is theoretically guaranteed to achieve the global optimum. In particular, we provide the closed-form solution for one challenging subproblem (i.e., the one with respect to the $\ell_{2,1}$ norm), such that both the subproblems can be efficiently and exactly solved. Note that LRR is a special case of our RKLRR when using the linear kernel. As a result, our algorithm is also suitable for solving the original LRR problem. In particular, it can well address the convergence issues of the existing solvers in [18] and [21] for solving the original LRR problem. In particular, the algorithm in [21] is not theoretically guaranteed to achieve the global optimum. Although the algorithm in [18] is proved to converge in theory, the total number of iterations of this method may

Manuscript received January 15, 2015; revised May 9, 2015 and July 21, 2015; accepted July 23, 2015. Date of publication September 29, 2015; date of current version October 17, 2016.

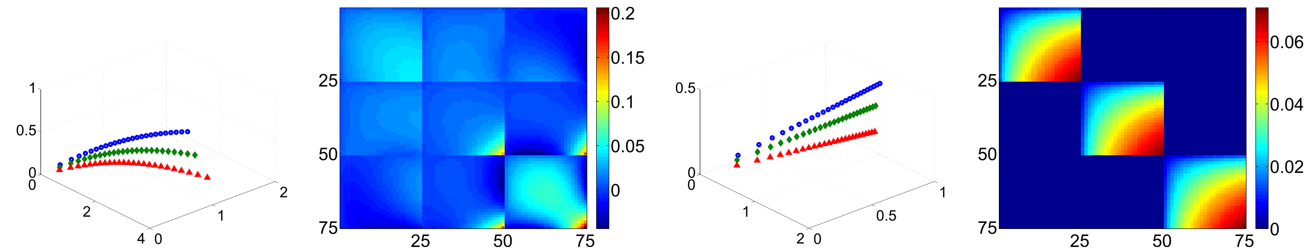
S. Xiao is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: xiaoshijiehithonor@gmail.com).

M. Tan is with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: mingkui.tan@adelaide.edu.au).

D. Xu and Z. Y. Dong are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: dongxudongxu@gmail.com; zydong@ieee.org).

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

Digital Object Identifier 10.1109/TNNLS.2015.2472284



(a) (b) (c) (d)

Fig. 1. Comparison between the original LRR and our proposed kernelized version of LRR for nonlinear subspace clustering. (a) Data $\{\mathbf{x}_i\}_{i=1}^n$ in the original input space. (b) Minimizer $\mathbf{Z}^*$ of the problem in (1). The corresponding *clustering error* (see Section VI for the definition) is ~30%. (c) Data $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ in the feature space. Here, we define $\phi(\mathbf{x})$ as $\phi(\mathbf{x}) = arctan(\mathbf{x})$, i.e., the *arctan* operation is performed on evey entry of $\mathbf{x}$. (d) Minimizer $\mathbf{Z}^*$ of the problem in (8). The corresponding *clustering error* is 0. This figure is best viewed in color.

be relatively large in practice (see Sections II and VI-A for more details).

Our major contributions are summarized as follows.

1) We first present the nonlinear version of LRR for handling clean data, and provide the closed-form solution for it. Moreover, we also propose RKLRR, which is a kernelized version of LRR to handle corrupted data. In addition, we show that many variants of LRR (such as nonnegative low rank and sparse (NNLRS) graph [15]) with the $\ell_{2,1}$ norm-based regularization on the representation error can be kernelized in a similar way.
2) Based on the ADM, we develop a new optimization algorithm to solve the optimization problem of RKLRR. By providing the analytical solution for the subproblem with respect to the $\ell_{2,1}$ norm, we efficiently solve both the resultant subproblems in closed form. In particular, our algorithm can be used to solve the original LRR problem. Our algorithm usually converges in less number of iterations, when compared with the existing LRR solvers in [18] and [21]. In addition, it is especially efficient when the feature dimension is high.
3) The experimental results on synthetic data sets demonstrate the efficiency of our algorithm for solving the original LRR problem, when compared with the existing LRR solvers in [18] and [21]. The comprehensive experimental results for different real-world tasks (i.e., face clustering and human activity clustering) clearly show the effectiveness of our RKLRR and the kernel versions of two variants of LRR.

The rest of this paper is organized as follows. We discuss the related works in Section II and introduce our formulations in Section III. In Section IV, our algorithm is presented, and its efficiency and the convergence property are also discussed. Moreover, we discuss the extensions of our kernelization technique to some variants of LRR in Section V. Finally, the experimental results are reported in Section VI, and the conclusion remarks are provided in Section VII.

II. Related Work

Notations: In the remainder of this paper, we use the lowercase/uppercase letter in boldface to denote a vector/matrix (e.g., $\mathbf{a}$ denotes a vector, and $\mathbf{A}$ denotes a matrix). The corresponding nonbold letter with a subscript denotes the entry in a vector/matrix (e.g., a_i denotes the ith entry of the vector $\mathbf{a}$, and A_{ij} denotes an entry at the ith row and the jth column of the matrix $\mathbf{A}$). The superscript $'$ denotes the transpose of a vector or a matrix. Let $\mathbf{I}_n$ denote the $n \times n$ identity matrix, and let $\mathbf{O}_{m\times n}$ denote the $m \times n$ zero matrix. Let $\mathbf{0}_n \in \mathbb{R}^n$ (respectively, $\mathbf{1}_n \in \mathbb{R}^n$) denote the column vector, where all the elements are zeros (ones). For simplicity, $\mathbf{I}$, $\mathbf{0}$, and $\mathbf{1}$ are used when the dimension is obvious.

Let $\|\mathbf{A}\|_*$ denote the nuclear norm [22], [23] of $\mathbf{A}$. Let $\mathrm{tr}(\mathbf{A})$ denote the trace of $\mathbf{A}$ (i.e., $\mathrm{tr}(\mathbf{A}) = \sum_i A_{i,i}$), and let $\langle \mathbf{A}, \mathbf{B}\rangle$ denote the inner product of $\mathbf{A}$ and $\mathbf{B}$ (i.e., $\langle \mathbf{A}, \mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}'\mathbf{B})$). $\|\mathbf{A}\|_{2,1} = \sum_{i=1}^n \|\mathbf{a}_i\|$ is the $\ell_{2,1}$ norm of $\mathbf{A}$, where $\mathbf{a}_i$ is the ith column of $\mathbf{A}$ and $\|\mathbf{a}\|$ is the ℓ_2 norm of $\mathbf{a}$. The max norm $\|\mathbf{A}\|_\infty$ of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_\infty = \max_{i,j} |A_{ij}|$. $\mathbf{A} \geq 0$ means that all the elements in $\mathbf{A}$ are nonnegative. For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, the inequality $\mathbf{a} \leq \mathbf{b}$ indicates that $a_i \leq b_i \forall i = 1, \ldots, n$. We use $\mathrm{diag}(\mathbf{a})$ or $\mathrm{diag}(\{a_i\}_{1\leq i\leq n})$ to denote a diagonal matrix, where $\mathbf{a} \in \mathbb{R}^n$ contains all the diagonal elements. Finally, let the operator $\circ$ denote the elementwise product (also called the Hadamard product) between two vectors, i.e., $\mathbf{a} \circ \mathbf{b}$ is a new vector with its ith element as $a_i b_i$.

First, we give a brief review of LRR. Suppose that we are given a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ containing n data points drawn from multiple subspaces, where d is the feature dimension. LRR seeks a low-rank representation $\mathbf{Z} \in \mathbb{R}^{n\times n}$ by minimizing $\|\mathbf{Z}\|_*$, subject to the constraint that the data are self-expressive (i.e., $\mathbf{X} = \mathbf{X}\mathbf{Z}$). Mathematically, for the LRR in the clean data case, the optimization problem [1] is as follows:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z}. \tag{1}$$

However, the real-world data often contain noise. To deal with grossly corrupted data, the following convex optimization problem is proposed for the LRR in the corrupted data case [1]:

$$\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{2,1}, \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E} \tag{2}$$

where λ is a positive tradeoff parameter, and $\|\mathbf{E}\|_{2,1}$ encourages the columnwise sparsity for the representation error $\mathbf{E} \in \mathbb{R}^{d\times n}$. In the real-world applications, such as face clustering, some data points may be corrupted (e.g., faces may be occluded by scarves), so the $\ell_{2,1}$ norm loss is particularly important for handling the outliers [1], [11], [21], and it is also widely used in many variants of LRR [15].

However, the optimization problem in (2) is difficult to be addressed by directly using the ADM [18]–[20]. To be exact, if we directly apply the ADM, the resultant optimization procedure will involve a nuclear-norm regularized optimization problem in the following form:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* + \frac{\nu}{2}\|\mathbf{XZ}-\mathbf{B}\|_F^2 \tag{3}$$

where both the scalar ν and the matrix $\mathbf{B}$ are irrelevant to $\mathbf{Z}$. However, it is a nontrivial task to solve the optimization problem (3). Therefore, the works in [18] and [21] attempt to address this issue using their own strategies.

In particular, Liu *et al.* [21] introduced an additional variable $\mathbf{J} = \mathbf{Z}$ into problem (2). Let us use the ADM with three blocks (ADM3B) to denote this algorithm. However, ADM3B is not theoretically guaranteed to obtain the global optimum [18], [24]. Moreover, it may require more memory and iterations, as mentioned in [18].

To address the convergence issue of ADM3B, Lin *et al.* [18] proposed the linearized ADM with adaptive penalty (LADMAP), without introducing one additional variable. In particular, to avoid solving the subproblem in the form of (3), the approximation is performed by replacing the squared Frobenius norm with a linear term plus a proximal term. LADMAP is theoretically proved to achieve the global optimum, under the assumption that all subproblems are exactly solved. However, possibly due to the approximation [18], the number of iterations is sometimes large when using LADMAP (see Section VI-A).

LRR is designed to handle the data from a union of subspaces in the original input space, so it may not effectively cope with the data sampled from nonlinear input spaces. Moreover, we cannot directly kernelize the optimization problem (2) for LRR, because the kernel trick cannot be readily used when the data do not only appear in the form of inner products [25] (i.e., $\mathbf{x}_i'\mathbf{x}_j$). While the work in [26] attempts to kernelize LRR, the proposed structural similarity and distance in learning (SSDL) method adopts the Frobenius norm to replace the $\ell_{2,1}$ norm for regularizing the representation error. Consequently, SSDL is not robust to outliers. Our experimental results for several real-world applications also demonstrate that our proposed RKLRR method generally outperforms SSDL (see Section VI). Recently, a fast LRR solver called FaLRR [27] was proposed by reformulating LRR with factorized data. However, the reformulation idea cannot be directly applied for many variants of LRR (e.g., NNLRS [15]) and their kernelized versions.

III. Formulations

Definitions: For the given data $\{\mathbf{x}_i\}_{i=1}^n$ in the input space, where $\mathbf{x}_i \in \mathbb{R}^d$, $\forall i = 1,\ldots,n$, we define $\mathbf{X} \in \mathbb{R}^{d\times n}$ as $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_n]$. Following the existing kernel-based methods, such as [28] and [29], let $\mathbf{K} \in \mathbb{R}^{n\times n}$ denote the kernel matrix, and let $ker(\mathbf{x},\mathbf{y})$ denote the kernel function. In particular, the (i,j)th element of $\mathbf{K}$, namely, K_{ij}, is calculated as follows:

$$K_{ij} = ker(\mathbf{x}_i,\mathbf{x}_j) \quad \forall i,j = 1,\ldots,n. \tag{4}$$

Without loss of generality, we assume that the kernel matrix is symmetric and positive semidefinite. According to [16], the kernel function $ker(\mathbf{x},\mathbf{y})$ induces a mapping ϕ: $\mathbb{R}^d \to \mathcal{F}$ (the new space $\mathcal{F}$ is referred to as the feature space [16]). Namely, for any $\mathbf{x},\mathbf{y} \in \mathbf{R}^d$, we have

$$ker(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})'\phi(\mathbf{y}). \tag{5}$$

Let us define $\Phi(\mathbf{X})$ as $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_n)]$, so that we have

$$\mathbf{K} = \Phi(\mathbf{X})'\Phi(\mathbf{X}). \tag{6}$$

Besides, let r_K denote the rank of $\mathbf{K}$, where $0 \le r_K \le n$. Considering that $\mathbf{K}$ is symmetric and positive semidefinite, we can use singular value decomposition (SVD) to decompose it in the form of

$$\mathbf{K} = \mathbf{V}\Sigma^2\mathbf{V}' \tag{7}$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix (i.e., $\mathbf{VV}' = \mathbf{V}'\mathbf{V} = \mathbf{I}_n$), and the diagonal matrix $\Sigma \in \mathbb{R}^{n\times n}$ is defined as $\Sigma = \mathrm{diag}([\sigma_1,\ldots,\sigma_{r_K},0,\ldots,0]')$, with the scalars $\{\sigma_i^2\}_{i=1}^{r_K}$ being the positive singular values of $\mathbf{K}$, which are sorted in descending order.

A. Clean Data Case

When the given data $\{\mathbf{x}_i\}_{i=1}^n$ are from multiple nonlinear subspaces, as shown in Fig. 1,[1] the minimizer $\mathbf{Z}^*$ for problem (1) may not well represent the relations between the given data. With the introduction of the kernel function which induces the mapping ϕ, $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ may lie in multiple linear subspaces on the new feature space, as illustrated in Fig. 1.

Based on the assumption that $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ resides in multiple linear subspaces, we propose the following nonlinear version of LRR in the clean data case:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_*, \quad \text{s.t. } \Phi(\mathbf{X}) = \Phi(\mathbf{X})\mathbf{Z}. \tag{8}$$

We show that problem (8) can be solved analytically and the closed-form solution is determined by the kernel matrix $\mathbf{K}$. In particular, we have the following theorem based on the above-mentioned definitions in (4)–(7).

Theorem 1: $\mathbf{V}_K\mathbf{V}_K'$ is the optimal solution of problem (8), where $\mathbf{V}_K \in \mathbb{R}^{n\times r_K}$ is obtained by using the first r_K columns of $\mathbf{V}$.

The proof of Theorem 1 is provided in Appendix A. As shown in Fig. 1, the membership between the given data is clearly discovered by the minimizer of (8).

B. Corrupted Data Case

We first reformulate problem (2) as a new constrained optimization problem. Based on the new formulation, we propose our RKLRR method. To be exact, different from [18] and [21], we first equivalently convert the problem in (2) into an unconstrained problem in (9). After that,

[1] The MATLAB code for generating the toy data is available at https://www.dropbox.com/s/5ol4p4mljzt6aob/GenerateToyData.m?dl=0.

we introduce a new variable and further convert the problem into a new constrained optimization problem (10), based on which we arrive at our RKLRR.

First, the equality constraint in (2) can be equivalently rewritten as $\mathbf{E} = \mathbf{X} - \mathbf{XZ}$. Therefore, the LRR problem (2) is equivalent to the following one:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* + \lambda \|\mathbf{X} - \mathbf{XZ}\|_{2,1}. \tag{9}$$

By defining a new variable $\mathbf{P} = \mathbf{I} - \mathbf{Z} \in \mathbb{R}^{n\times n}$, we can rewrite $\|\mathbf{X} - \mathbf{XZ}\|_{2,1}$ in (9) as $\|\mathbf{X} - \mathbf{XZ}\|_{2,1} = \|\mathbf{XP}\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{Xp}_i\| = \sum_{i=1}^{n} (\mathbf{p}_i'\mathbf{X}'\mathbf{Xp}_i)^{1/2}$, where $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_n]$, and $\mathbf{p}_i \in \mathbb{R}^n$ denotes the ith column of $\mathbf{P}$, $\forall i = 1, \ldots, n$. Accordingly, the problem in (9) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{P}} \ \|\mathbf{Z}\|_* + \lambda \sum_{i=1}^{n} \sqrt{\mathbf{p}_i'\mathbf{X}'\mathbf{Xp}_i} \\ &\text{s.t.} \ \ \mathbf{P} = \mathbf{I} - \mathbf{Z}. \end{aligned} \tag{10}$$

Apparently, after converting the LRR problem in (2) into the formulation in (10), the data $\{\mathbf{x}_i\}_{i=1}^n$ now only appear in the form of inner product (i.e., $\mathbf{x}_i'\mathbf{x}_j$), in the term $\mathbf{X}'\mathbf{X}$. As a result, the kernelization can be easily performed.

In particular, we arrive at the following nonlinear version of LRR by replacing $\mathbf{X}$ in (9) with $\Phi(\mathbf{X})$:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* + \lambda \|\Phi(\mathbf{X}) - \Phi(\mathbf{X})\mathbf{Z}\|_{2,1}. \tag{11}$$

By defining a function $g(\mathbf{P})$ with respect to $\mathbf{P} \in \mathbb{R}^{n\times n}$ as

$$g(\mathbf{P}) \triangleq \sum_{i=1}^{n} \sqrt{\mathbf{p}_i'\mathbf{Kp}_i} \tag{12}$$

where $\mathbf{p}_i \in \mathbb{R}^n$ denotes the ith column of $\mathbf{P}$, $\forall i = 1, \ldots, n$, we can equivalently rewrite (11) as the following RKLRR problem:

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{P}} \ \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) \\ &\text{s.t.} \ \mathbf{P} = \mathbf{I} - \mathbf{Z} \end{aligned} \tag{13}$$

where the kernel matrix $\mathbf{K}$ is contained in $g(\mathbf{P})$. Note that, when using the linear kernel $\mathbf{K} = \mathbf{X}'\mathbf{X}$, the RKLRR problem in (13) is reduced to the form in (10), so LRR is a special case of RKLRR when using the linear kernel.

IV. Optimization

In this paper, we propose to solve problem (13) using the ADM [18]–[20] with two blocks of variables (namely, $\mathbf{Z}$ and $\mathbf{P}$). In particular, we introduce the Lagrange multiplier $\mathbf{L} \in \mathbb{R}^{n\times n}$ and operate on the augmented Lagrange function $\mathcal{L}(\mathbf{Z}, \mathbf{P}, \mathbf{L})$ as follows:

$$\begin{aligned} &\mathcal{L}(\mathbf{Z}, \mathbf{P}, \mathbf{L}) \\ &\quad = \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) + \langle \mathbf{L}, \mathbf{I} - \mathbf{Z} - \mathbf{P}\rangle + \frac{\rho}{2}\|\mathbf{I} - \mathbf{Z} - \mathbf{P}\|_F^2 \\ &\quad = \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) + \frac{\rho}{2}\left\|\mathbf{I} - \mathbf{Z} - \mathbf{P} + \frac{\mathbf{L}}{\rho}\right\|_F^2 - \frac{\|\mathbf{L}\|_F^2}{2\rho} \end{aligned}$$

where ρ is a positive penalty parameter.

Algorithm 1 Proposed Algorithm for Solving RKLRR

```
Input: $\lambda$, the SVD of $\mathbf{K}$.
Initialize $\mathbf{P}_0$ as $\mathbf{I}$, and initialize $\mathbf{Z}_0, \mathbf{L}_0$ as $\mathbf{O}_{n\times n}$.
$(\rho_0, \rho_{max}, \Delta\rho, \epsilon, N_{iter}) \leftarrow (0.5, 10^6, 0.1, 10^{-5}, 10^6)$.
for $t = 0 : N_{iter}$ do
  1. $\mathbf{Z}_{t+1} \leftarrow \arg\min_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{P}_t, \mathbf{L}_t)$.
  2. $\mathbf{P}_{t+1} \leftarrow \arg\min_{\mathbf{P}} \mathcal{L}(\mathbf{Z}_{t+1}, \mathbf{P}, \mathbf{L}_t)$.
  3. $\mathbf{L}_{t+1} \leftarrow \mathbf{L}_t + \rho_t(\mathbf{I} - \mathbf{Z}_{t+1} - \mathbf{P}_{t+1})$.
  4. $\rho_{t+1} \leftarrow \min(\rho_t(1 + \Delta\rho), \rho_{max})$.
  5. Break if $\|\mathbf{I} - \mathbf{Z}_{t+1} - \mathbf{P}_{t+1}\|_\infty \leq \epsilon$.
end for
Output: the data representation $\mathbf{Z}^* = \mathbf{Z}_{t+1}$.
```

The details of our algorithm for solving the RKLRR are summarized in Algorithm 1. As shown in Algorithm 1, at the tth iteration, we alternatively update $\mathbf{Z}_{t+1}$, $\mathbf{P}_{t+1}$, $\mathbf{L}_{t+1}$, and ρ_{t+1} until the convergence condition is satisfied. The updating steps for $\mathbf{L}_{t+1}$ and ρ_{t+1} are similar to those in [1], as shown in Algorithm 1 (see the third and fourth steps in the loop). Therefore, we only detail the first two steps for updating $\mathbf{Z}_{t+1}$ and $\mathbf{P}_{t+1}$, which are also the main steps at each iteration. In particular, it is more challenging to update $\mathbf{P}_{t+1}$.

In particular, the optimization problem for updating $\mathbf{Z}_{t+1}$, i.e., $\min_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{P}_t, \mathbf{L}_t)$, can be detailed as follows:

$$\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* + \frac{\rho_t}{2}\|\mathbf{Z} - (\mathbf{I} - \mathbf{P}_t + \mathbf{L}_t/\rho_t)\|_F^2 \tag{14}$$

in which we have dropped the terms that are irrelevant to $\mathbf{Z}$. This subproblem can be solved in closed-form using the singular value shrinkage operator [30, Th. 2.1].

Moreover, the optimization problem for updating $\mathbf{P}_{t+1}$, i.e., $\min_{\mathbf{P}} \mathcal{L}(\mathbf{Z}_{t+1}, \mathbf{P}, \mathbf{L}_t)$, can be written as follows:

$$\min_{\mathbf{P}} \ \lambda g(\mathbf{P}) + \frac{\rho_t}{2}\|\mathbf{P} - \mathbf{C}_{t+1}\|_F^2 \tag{15}$$

where $\mathbf{C}_{t+1} = \mathbf{I} - \mathbf{Z}_{t+1} + \mathbf{L}_t/\rho_t$, and we have dropped the terms that are irrelevant to $\mathbf{P}$.

Note that it is nontrivial to solve the problem in (15), where the objective function is convex but nonsmooth. Nevertheless, in this paper, we show that it can be solved analytically.

Hereafter, we drop the subscripts in $\mathbf{C}_{t+1}$ for ease of presentation. Moreover, by defining $\mathbf{c}_i$ (respectively, $\mathbf{p}_i$) as the ith column of $\mathbf{C}$ (respectively, $\mathbf{P}$) and scalar τ as $\tau \triangleq (\rho_t/\lambda)$, we have $(\rho_t/2)\|\mathbf{P} - \mathbf{C}\|_F^2 = \lambda((\tau/2)\sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{c}_i\|^2)$. After dividing the objective by λ, we can equivalently rewrite the problem in (15) as follows:

$$\min_{\{\mathbf{p}_i\}_{i=1}^n} \ \sum_{i=1}^{n} \sqrt{\mathbf{p}_i'\mathbf{Kp}_i} + \frac{\tau}{2}\sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{c}_i\|^2. \tag{16}$$

Note that the optimization problem (16) is separable with respect to $\mathbf{p}_i$'s, so it can be decomposed into n subproblems, and each subproblem is in the following form:

$$\min_{\mathbf{p}_i} \ \sqrt{\mathbf{p}_i'\mathbf{Kp}_i} + \frac{\tau}{2}\|\mathbf{p}_i - \mathbf{c}_i\|^2. \tag{17}$$

Before introducing the analytical solution of (17) in Theorem 2, we first provide the following lemma.

Lemma 1: Given a positive scalar τ, a constant vector $\mathbf{h} \in \mathbb{R}^q$, with q being a positive integer, and a diagonal matrix $\mathbf{S} = \mathrm{diag}(\{s_i\}_{1\le i\le q}) \in \mathbb{R}^{q\times q}$, with $\{s_i\}_{i=1}^q$ being positive scalars sorted in descending order, the optimal solution $\mathbf{p}^*$ for the following problem:

$$\min_{\mathbf{p}\in\mathbb{R}^q} \sqrt{\mathbf{p}'\mathbf{S}^2\mathbf{p}} + \frac{\tau}{2}\|\mathbf{p}-\mathbf{h}\|^2 \tag{18}$$

is given by

$$\mathbf{p}^* = \begin{cases} \left(\frac{\mathbf{S}^2}{\tau\alpha} + \mathbf{I}\right)^{-1}\mathbf{h}, & \text{if } \|\mathbf{S}^{-1}\mathbf{h}\| > \frac{1}{\tau} \\ \mathbf{0}_q, & \text{otherwise} \end{cases} \tag{19}$$

where $\mathbf{S}^{-1} = \mathrm{diag}(\{s_i^{-1}\}_{1\le i\le q})$, and $\alpha > 0$ satisfies

$$\mathbf{h}'\mathrm{diag}\left(\left\{\frac{s_i^2}{(\tau\alpha + s_i^2)^2}\right\}_{1\le i\le q}\right)\mathbf{h} = \frac{1}{\tau^2}. \tag{20}$$

In particular, when $\|\mathbf{S}^{-1}\mathbf{h}\| > 1/\tau$, (20) (with respect to α) has exactly one positive root, which can be obtained by the bisection method [31].

The proof of Lemma 1 is provided in Appendix B. Based on Lemma 1, we have the following theorem.

Theorem 2: The optimal solution $\mathbf{p}_i^*$ of problem (17) (where $\tau > 0$) is

$$\mathbf{p}_i^* = \begin{cases} \hat{\mathbf{p}}, & \text{if } \|\,[\frac{1}{\sigma_1},\ldots,\frac{1}{\sigma_{r_K}}]' \circ \tilde{\mathbf{h}}_u\| > \frac{1}{\tau} \\ \mathbf{c}_i - \mathbf{V}_K\tilde{\mathbf{h}}_u, & \text{otherwise} \end{cases} \tag{21}$$

in which $\tilde{\mathbf{h}}_u = \mathbf{V}_K'\mathbf{c}_i \in \mathbb{R}^{r_K}$, where $\mathbf{V}_K \in \mathbb{R}^{n\times r_K}$ is formed by the first r_K columns of $\mathbf{V}$, and the vector $\hat{\mathbf{p}}$ is defined as

$$\hat{\mathbf{p}} = \mathbf{c}_i - \mathbf{V}_K\left(\left[\frac{\sigma_1^2}{\tau\alpha+\sigma_1^2},\ldots,\frac{\sigma_{r_K}^2}{\tau\alpha+\sigma_{r_K}^2}\right]' \circ \tilde{\mathbf{h}}_u\right) \tag{22}$$

where α is a positive scalar, satisfying

$$\tilde{\mathbf{h}}_u'\mathrm{diag}\left(\left\{\frac{\sigma_i^2}{(\tau\alpha+\sigma_i^2)^2}\right\}_{1\le i\le r_K}\right)\tilde{\mathbf{h}}_u = \frac{1}{\tau^2}. \tag{23}$$

In particular, when $\|[1/\sigma_1,\ldots,1/\sigma_{r_K}]' \circ \tilde{\mathbf{h}}_u\| > 1/\tau$, the equation in (23) (with respect to α) has a unique positive root, which can be obtained by the bisection method [31].

Proof: By substituting $\mathbf{K} = \mathbf{V}\Sigma^2\mathbf{V}'$ into $(\mathbf{p}_i'\mathbf{K}\mathbf{p}_i)^{1/2}$ [i.e., the first term in the objective function of (17)], we arrive at

$$\sqrt{\mathbf{p}_i'\mathbf{K}\mathbf{p}_i} = \sqrt{\mathbf{p}_i'\mathbf{V}\Sigma^2\mathbf{V}'\mathbf{p}_i} = \sqrt{(\mathbf{V}'\mathbf{p}_i)'\Sigma^2(\mathbf{V}'\mathbf{p}_i)}. \tag{24}$$

On the other hand, since $\mathbf{V}\mathbf{V}' = \mathbf{I}$, we have

$$\|\mathbf{p}_i - \mathbf{c}_i\|^2 = \|\mathbf{V}'(\mathbf{p}_i - \mathbf{c}_i)\|^2 = \|\mathbf{V}'\mathbf{p}_i - \mathbf{V}'\mathbf{c}_i\|^2. \tag{25}$$

Note that, in (24) and (25), the variable $\mathbf{p}_i$ only appears in $\mathbf{V}'\mathbf{p}_i$, and the term $\mathbf{V}'\mathbf{c}_i$ in (25) is constant with respect to $\mathbf{p}_i$. Therefore, by defining a new variable $\tilde{\mathbf{p}} = \mathbf{V}'\mathbf{p}_i$ and a constant vector

$$\tilde{\mathbf{h}} = \mathbf{V}'\mathbf{c}_i \tag{26}$$

we reformulate the optimization problem (17) as the following equivalent one:

$$\min_{\tilde{\mathbf{p}}} \sqrt{\tilde{\mathbf{p}}'\Sigma^2\tilde{\mathbf{p}}} + \frac{\tau}{2}\|\tilde{\mathbf{p}} - \tilde{\mathbf{h}}\|^2. \tag{27}$$

Let $\tilde{\mathbf{p}}^*$ denote the optimal solution of the problem in (27). Since $\mathbf{V}\mathbf{V}' = \mathbf{I}$, we have $\mathbf{p}_i = \mathbf{V}\tilde{\mathbf{p}}$ for any $\tilde{\mathbf{p}} = \mathbf{V}'\mathbf{p}_i$. Therefore, once we obtain $\tilde{\mathbf{p}}^*$, the optimal solution $\mathbf{p}_i^*$ of the problem in (17) can be obtained by

$$\mathbf{p}_i^* = \mathbf{V}\tilde{\mathbf{p}}^*. \tag{28}$$

Now, let us discuss how to obtain $\tilde{\mathbf{p}}^*$. Note that Σ is a diagonal matrix, where the last $n - r_K$ diagonal elements are all zeros, so we express the variable $\tilde{\mathbf{p}} \in \mathbb{R}^n$ as $\tilde{\mathbf{p}} = [\tilde{\mathbf{p}}_u', \tilde{\mathbf{p}}_d']'$, where $\tilde{\mathbf{p}}_u \in \mathbb{R}^{r_K}$ and $\tilde{\mathbf{p}}_d \in \mathbb{R}^{(n-r_K)}$. Similarly, let us split $\mathbf{h} \in \mathbb{R}^n$ into two vectors $\tilde{\mathbf{h}}_u \in \mathbb{R}^{r_K}$ and $\tilde{\mathbf{h}}_d \in \mathbb{R}^{(n-r_K)}$, that is

$$\tilde{\mathbf{h}} = [\tilde{\mathbf{h}}_u', \tilde{\mathbf{h}}_d']'. \tag{29}$$

According to (26), we can write $\tilde{\mathbf{h}}_u$ and $\tilde{\mathbf{h}}_d$ as follows:

$$\tilde{\mathbf{h}}_u = [\mathbf{I}_{r_K}, \mathbf{O}_{r_K\times(n-r_K)}]\mathbf{V}'\mathbf{c}_i = \mathbf{V}_K'\mathbf{c}_i \tag{30}$$

$$\tilde{\mathbf{h}}_d = [\mathbf{O}_{(n-r_K)\times r_K}, \mathbf{I}_{(n-r_K)}]\mathbf{V}'\mathbf{c}_i. \tag{31}$$

Moreover, we define a diagonal matrix $\Sigma_u \in \mathbb{R}^{r_K\times r_K}$ as

$$\Sigma_u = \mathrm{diag}(\{\sigma_i\}_{1\le i\le r_K}). \tag{32}$$

With these definitions, we equivalently rewrite (27) as

$$\min_{\tilde{\mathbf{p}}_u,\tilde{\mathbf{p}}_d}\left(\sqrt{\tilde{\mathbf{p}}_u'\Sigma_u{}^2\tilde{\mathbf{p}}_u} + \frac{\tau}{2}\|\tilde{\mathbf{p}}_u - \tilde{\mathbf{h}}_u\|^2\right) + \left(\frac{\tau}{2}\|\tilde{\mathbf{p}}_d - \tilde{\mathbf{h}}_d\|^2\right)$$

where the objective function is separable with respect to $\tilde{\mathbf{p}}_u$ and $\tilde{\mathbf{p}}_d$. Let $(\tilde{\mathbf{p}}_u^*, \tilde{\mathbf{p}}_d^*)$ denote the optimal solution to the above-mentioned optimization problem. Accordingly, $\tilde{\mathbf{p}}^*$ can be written as

$$\tilde{\mathbf{p}}^* = [(\tilde{\mathbf{p}}_u^*)', (\tilde{\mathbf{p}}_d^*)']'. \tag{33}$$

It is obvious that

$$\tilde{\mathbf{p}}_d^* = \tilde{\mathbf{h}}_d \tag{34}$$

and $\tilde{\mathbf{p}}_u^*$ is an optimal solution to the following problem:

$$\min_{\tilde{\mathbf{p}}_u} \sqrt{\tilde{\mathbf{p}}_u'\Sigma_u{}^2\tilde{\mathbf{p}}_u} + \frac{\tau}{2}\|\tilde{\mathbf{p}}_u - \tilde{\mathbf{h}}_u\|^2. \tag{35}$$

Note that (35) is exactly in the form of (18) in Lemma 1. According to Lemma 1, $\tilde{\mathbf{p}}_u^*$ is given by

$$\tilde{\mathbf{p}}_u^* = \begin{cases} \left(\frac{\Sigma_u{}^2}{\tau\alpha} + \mathbf{I}\right)^{-1}\tilde{\mathbf{h}}_u, & \text{if } \|\Sigma_u^{-1}\tilde{\mathbf{h}}_u\| > \frac{1}{\tau} \\ \mathbf{0}_{r_K}, & \text{otherwise} \end{cases} \tag{36}$$

where α is a positive scalar, satisfying the equation in (23). In particular, when $\|\Sigma_u^{-1}\tilde{\mathbf{h}}_u\| > 1/\tau$, the equation in (23) (with respect to α) has exactly one positive root, which can be obtained by the bisection method [31].

Given (28), (33), and (34), we have $\mathbf{p}_i^* = \mathbf{V}[(\tilde{\mathbf{p}}_u^*)', \tilde{\mathbf{h}}_d']'$. By substituting (36) into the equation, we arrive at

$$\mathbf{p}_i^* = \begin{cases} \mathbf{V}\begin{bmatrix} \left(\frac{{\Sigma_u}^2}{\tau\alpha} + \mathbf{I}\right)^{-1} \tilde{\mathbf{h}}_u \\ \tilde{\mathbf{h}}_d \end{bmatrix}, & \text{if } \|\Sigma_u^{-1}\tilde{\mathbf{h}}_u\| > \frac{1}{\tau} \\ \mathbf{V}\begin{bmatrix} \mathbf{0}_{r_K} \\ \tilde{\mathbf{h}}_d \end{bmatrix}, & \text{otherwise.} \end{cases} \tag{37}$$

Now, to complete the proof of Theorem 2, the remaining tasks are to verify the following three equalities:

$$\Sigma_u^{-1}\tilde{\mathbf{h}}_u = \left[\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_{r_K}}\right]' \circ \tilde{\mathbf{h}}_u \tag{38}$$

$$\mathbf{V}\begin{bmatrix} \left(\frac{{\Sigma_u}^2}{\tau\alpha} + \mathbf{I}\right)^{-1} \tilde{\mathbf{h}}_u \\ \tilde{\mathbf{h}}_d \end{bmatrix} = \hat{\mathbf{p}} \tag{39}$$

$$\mathbf{V}\begin{bmatrix} \mathbf{0}_{r_K} \\ \tilde{\mathbf{h}}_d \end{bmatrix} = \mathbf{c}_i - \mathbf{V}_K\tilde{\mathbf{h}}_u. \tag{40}$$

Let us verify them as follows. First, the equality in (38) can be easily verified based on the definition of Σ_u in (32).

Besides, we can prove (39) and (40) based on the following property. For any $\mathbf{a} = [a_1, \ldots, a_{r_K}]' \in \mathbb{R}^{r_K}$, we have

$$\begin{aligned} &\mathbf{V}\text{diag}([\mathbf{a}', \mathbf{1}_{n-r_K}']')\mathbf{V}'\mathbf{c}_i \\ &\quad = \mathbf{c}_i - \mathbf{V}_K\text{diag}(\{1 - a_i\}_{1 \le i \le r_K})\mathbf{V}_K'\mathbf{c}_i \qquad (41) \\ &\quad = \mathbf{c}_i - \mathbf{V}_K\text{diag}(\{1 - a_i\}_{1 \le i \le r_K})\tilde{\mathbf{h}}_u \qquad (42) \end{aligned}$$

where the equality in (41) can be verified, since $\mathbf{V}\mathbf{V}' = \mathbf{I}_n$ and $[\mathbf{a}', \mathbf{1}_{n-r_K}']' = \mathbf{1}_n - [(\mathbf{1}_{r_K} - \mathbf{a})', \mathbf{0}_{n-r_K}']'$, while the equality in (42) holds based on (30) and the definition of $\mathbf{V}_K$.

In fact, (39) can be obtained based on the above-mentioned property, by replacing $\mathbf{a}$ with $[\tau\alpha/(\tau\alpha + \sigma_1^2), \ldots, \tau\alpha/(\tau\alpha + \sigma_{r_K}^2)]'$. In particular, for the equality in (39), its left-hand side can be rewritten as $\mathbf{V}\text{diag}([[\tau\alpha/(\tau\alpha + \sigma_1^2), \ldots, \tau\alpha/(\tau\alpha + \sigma_{r_K}^2)]', \mathbf{1}_{n-r_K}']')\mathbf{V}'\mathbf{c}_i$, based on the equality $({\Sigma_u}^2/(\tau\alpha) + \mathbf{I})^{-1} = \text{diag}(\{\tau\alpha/(\tau\alpha + \sigma_i^2)\}_{1 \le i \le r_K})$ and the equalities in (30) and (31). Moreover, the right-hand side of the equality in (39), namely, $\hat{\mathbf{p}}$ defined in (22), can be written as $\mathbf{c}_i - \mathbf{V}_K\text{diag}(\{1 - (\tau\alpha)/(\tau\alpha + \sigma_i^2)\}_{1 \le i \le r_K})\tilde{\mathbf{h}}_u$. Based on the above-mentioned property, we can conclude that $\mathbf{V}\text{diag}([[\tau\alpha/(\tau\alpha + \sigma_1^2), \ldots, \tau\alpha/(\tau\alpha + \sigma_{r_K}^2)]', \mathbf{1}_{n-r_K}']')\mathbf{V}'\mathbf{c}_i$ is equal to $\mathbf{c}_i - \mathbf{V}_K\text{diag}(\{1 - (\tau\alpha)/(\tau\alpha + \sigma_i^2)\}_{1 \le i \le r_K})\tilde{\mathbf{h}}_u$, so that (39) is obtained.

We can also obtain (40) based on the above-mentioned property, by replacing $\mathbf{a}$ with $\mathbf{0}_{r_K}$. In particular, the left-hand side of (40) is equal to $\mathbf{V}\text{diag}([\mathbf{0}_{r_K}', \mathbf{1}_{n-r_K}']')\mathbf{V}'\mathbf{c}_i$ based on (31), which can be rewritten as $\mathbf{c}_i - \mathbf{V}_K\text{diag}(\mathbf{1}_{r_K} - \mathbf{0}_{r_K})\tilde{\mathbf{h}}_u$ based on the above-mentioned property. Note that we can easily rewrite $\mathbf{c}_i - \mathbf{V}_K\text{diag}(\mathbf{1}_{r_K} - \mathbf{0}_{r_K})\tilde{\mathbf{h}}_u$ as the right-hand side of (40), so (40) is obtained.

This completes the proof of Theorem 2. □

Now, let us discuss the efficiency and the convergence of Algorithm 1.

A. Efficiency

It is worth mentioning that, when the linear kernel $\mathbf{K} = \mathbf{X}'\mathbf{X}$ is used, Algorithm 1 can be readily used to solve the original LRR problem. In particular, when using Algorithm 1 to solve the original LRR problem, we can directly obtain the SVD of the linear kernel $\mathbf{K} = \mathbf{X}'\mathbf{X}$ based on the SVD of $\mathbf{X}$, without calculating $\mathbf{K}$ explicitly. In particular, after calculating the SVD of $\mathbf{X}$ (i.e., $\mathbf{X} = \mathbf{U}_X\mathbf{S}_X\mathbf{V}_X'$) with $O(ndl)$ time complexity, where $l = \min(n, d)$, we can readily obtain the SVD of $\mathbf{K}$ as $\mathbf{K} = \mathbf{V}_X\mathbf{S}_X^2\mathbf{V}_X'$. Note that $\mathbf{S}_X^2$ can be simply calculated in an elementwise way, because $\mathbf{S}_X$ is a diagonal matrix.

Now, let us discuss the efficiency of our Algorithm 1, as well as the existing algorithms LADMAP and ADM3B, for solving the original LRR problem. Generally speaking, the efficiency of an iterative algorithm is determined by the total number of iterations, as well as the time complexity at each iteration. For the total number of iterations, we experimentally demonstrate that our Algorithm 1 usually converges after fewer number of iterations than the two existing algorithms ADM3B [21] and LADMAP [18] (see Section VI-A for more details).

Regarding the computational complexity at each iteration, we need to update $\mathbf{Z}_{t+1}$ and $\mathbf{P}_{t+1}$, which leads to the main cost of Algorithm 1. When updating $\mathbf{Z}_{t+1}$, one can perform singular value shrinkage [30] through partial SVD similarly as in [18], with $O(rn^2)$ time complexity, where r is the rank for partial SVD at an iteration. When updating $\mathbf{P}_{t+1}$, we need to solve n subproblems in the form of (17), where the time complexity of each subproblem is $O(r_X n)$ due to the matrix–vector multiplication in Theorem 2, where r_X is the rank of $\mathbf{X}$. Therefore, the complexity of this process is $O(r_X n^2)$. Thus, at each iteration, the complexity of Algorithm 1 is $O((r_X + r)n^2)$. In contrast, the time complexity of LADMAP at each iteration is $O(\hat{r}nd + \hat{r}n^2)$, where $\hat{r}$ is the predicted rank of $\mathbf{Z}$ at an iteration of LADMAP. The time complexity of ADM3B at each iteration is at most $O(ndl + nl^2)$ [21], where $l = min(n, d)$. Nevertheless, ADM3B is not theoretically guaranteed to obtain the global optimum [18], [24]. Note that the complexity of our Algorithm 1 is irrelevant to d as long as the SVD of $\mathbf{K}$ (or $\mathbf{X}$) is given, so it is especially suitable for handling high-dimensional data with $d \gg n$.

B. Convergence

Based on [18]–[20], we have the following theorem regarding the convergence of Algorithm 1.

Theorem 3: The sequence $\{(\mathbf{Z}_t, \mathbf{P}_t, \mathbf{L}_t)\}$ generated by Algorithm 1 converges to an accumulation point. In particular, we have $\|\mathbf{Z}_{t+1} - \mathbf{Z}_t\|_F \to 0$, $\|\mathbf{P}_{t+1} - \mathbf{P}_t\|_F \to 0$, and $\|\mathbf{L}_{t+1} - \mathbf{L}_t\|_F \to 0$. Moreover, the accumulation point is an optimal solution of the optimization problem (13).

In contrast, ADM3B [1] is not theoretically guaranteed to achieve the global optimum. Regarding LADMAP [18], although it is theoretically proved to achieve the global optimum based on the assumption that its subproblems are solved exactly, its total number of iterations may be larger than that of our Algorithm 1 (see Section VI-A for details).

Algorithm 2 Algorithm for Solving RKNNLRS

```
Input: $\lambda, \beta$, the SVD of $\mathbf{K}$.
Initialize $\mathbf{P}_0$ as $\mathbf{I}$ and initialize $\mathbf{Z}_0, \mathbf{W}_0, \mathbf{L}_0, \mathbf{U}_0$ as $\mathbf{O}_{n\times n}$.
$(\rho_0, \rho_{max}, \Delta\rho, \epsilon, N_{iter}) \leftarrow (0.5, 10^6, 0.1, 10^{-5}, 10^6)$.
for $t = 0 : N_{iter}$ do
  1. $\mathbf{Z}_{t+1} \leftarrow \arg\min_{\mathbf{Z}} \mathcal{L}(\mathbf{Z}, \mathbf{P}_t, \mathbf{W}_t, \mathbf{L}_t, \mathbf{U}_t)$.
  2. $\mathbf{P}_{t+1} \leftarrow \arg\min_{\mathbf{P}} \mathcal{L}(\mathbf{Z}_{t+1}, \mathbf{P}, \mathbf{W}_t, \mathbf{L}_t, \mathbf{U}_t)$.
  3. $\mathbf{W}_{t+1} \leftarrow \mathcal{S}_{\beta/\rho_t}(\mathbf{Z}_{t+1} - \frac{\mathbf{U}_t}{\rho_t})$, and set the negative elements in $\mathbf{W}_{t+1}$ to zeros.
  4. $\mathbf{L}_{t+1} \leftarrow \mathbf{L}_t + \rho_t(\mathbf{I} - \mathbf{Z}_{t+1} - \mathbf{P}_{t+1})$,
     $\mathbf{U}_{t+1} \leftarrow \mathbf{U}_t + \rho_t(\mathbf{W}_{t+1} - \mathbf{Z}_{t+1})$.
  5. $\rho_{t+1} \leftarrow \min(\rho_t(1 + \Delta\rho), \rho_{max})$.
  6. Break if the conditions $\|\mathbf{I} - \mathbf{Z}_{t+1} - \mathbf{P}_{t+1}\|_\infty \le \epsilon$ and
     $\|\mathbf{W}_{t+1} - \mathbf{Z}_{t+1}\|_\infty \le \epsilon$ are both satisfied.
end for
Output: the data representation $\mathbf{Z}^* = \mathbf{Z}_{t+1}$.
```

V. Extensions

Generally speaking, our reformulation and kernelization presented in Section III, as well as the optimization techniques proposed in Section IV, can be used to extend many existing variants of LRR [15] with the $\ell_{2,1}$ norm-based regularization on the representation error. In this paper, we take an NNLRS graph [15] as an example. NNLRS extends LRR [1] by additionally encouraging $\mathbf{Z}$ to be sparse and nonnegative as

$$\begin{aligned} &\min_{\mathbf{Z}\ge 0,\mathbf{E}} \ \|\mathbf{Z}\|_* + \lambda\|\mathbf{E}\|_{2,1} + \beta\|\mathbf{Z}\|_1 \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E} \end{aligned} \tag{43}$$

where β is a positive tradeoff parameter. By similarly performing our reformulation and kernelization, as described in Section III, we arrive at the following robust kernel NNLRS (RKNNLRS) graph problem:

$$\begin{aligned} &\min_{\mathbf{Z}\ge 0,\mathbf{P}} \ \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) + \beta\|\mathbf{Z}\|_1 \\ &\text{s.t. } \mathbf{P} = \mathbf{I} - \mathbf{Z}. \end{aligned} \tag{44}$$

where $g(\mathbf{P})$ contains the kernel matrix $\mathbf{K} = \Phi(\mathbf{X})'\Phi(\mathbf{X})$. To solve problem (44), we introduce $\mathbf{W} = \mathbf{Z} \in \mathbb{R}^{n\times n}$, so the optimization problem becomes

$$\begin{aligned} &\min_{\mathbf{W}\ge 0,\mathbf{Z},\mathbf{P}} \ \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) + \beta\|\mathbf{W}\|_1 \\ &\text{s.t. } \mathbf{P} = \mathbf{I} - \mathbf{Z}, \ \ \mathbf{W} = \mathbf{Z}. \end{aligned}$$

Accordingly, the augmented Lagrange function is as follows:

$$\begin{aligned} &\mathcal{L}(\mathbf{Z}, \mathbf{P}, \mathbf{W}, \mathbf{L}, \mathbf{U}) \\ &\quad = \|\mathbf{Z}\|_* + \lambda g(\mathbf{P}) + \beta\|\mathbf{W}\|_1 + \langle \mathbf{I} - \mathbf{Z} - \mathbf{P}, \mathbf{L}\rangle \\ &\qquad + \langle \mathbf{W} - \mathbf{Z}, \mathbf{U}\rangle + \frac{\rho}{2}\left(\|\mathbf{I} - \mathbf{Z} - \mathbf{P}\|_F^2 + \|\mathbf{W} - \mathbf{Z}\|_F^2\right) \end{aligned}$$

where $\mathbf{L}$ and $\mathbf{U}$ are the Lagrange multipliers, and ρ is the penalty parameter.

The corresponding optimization is shown in Algorithm 2. In particular, similar to the one in (14), the subproblem for updating $\mathbf{Z}_{t+1}$ is solved in closed-form according to [30]. Moreover, the subproblem for updating $\mathbf{P}_{t+1}$ is solved in closed-form according to Theorem 2. Finally, the subproblem for updating $\mathbf{W}_{t+1}$ is $\min_{\mathbf{W}\ge 0} \mathcal{L}(\mathbf{Z}_{t+1}, \mathbf{P}_{t+1}, \mathbf{W}, \mathbf{L}_t, \mathbf{U}_t)$, namely, $\min_{\mathbf{W}\ge 0} \ \beta\|\mathbf{W}\|_1 + (\rho_t/2)\|\mathbf{W} - (\mathbf{Z}_{t+1} - (\mathbf{U}_t/\rho_t))\|_F^2$. Based on [15], the optimal solution can be obtained by setting the negative elements (if any) in $\mathcal{S}_{\beta/\rho_t}(\mathbf{Z}_{t+1} - (\mathbf{U}_t/\rho_t))$ to zeros, where $\mathcal{S}_{\cdot}(\cdot)$ is the soft-thresholding (shrinkage) operator [15].

In this paper, we also implement a simpler version of NNLRS, which is referred to as LRS, by dropping the nonnegative constraint on $\mathbf{Z}$ (i.e., $\mathbf{Z} \ge 0$) in (43). Moreover, the corresponding kernelized version of LRS (referred to as RKLRS) is the problem in (44) after removing the constraint $\mathbf{Z} \ge 0$. RKLRS can be solved using Algorithm 2 with a simple adjustment, i.e., skipping the process "set the negative elements in $\mathbf{W}_{t+1}$ to zeros" in the third step of each iteration. In fact, when solving RKLRS, the subproblem for updating $\mathbf{W}_{t+1}$ is $\min_{\mathbf{W}} \mathcal{L}(\mathbf{Z}_{t+1}, \mathbf{P}_{t+1}, \mathbf{W}, \mathbf{L}_t, \mathbf{U}_t)$, which can be detailed as $\min_{\mathbf{W}} \ \beta\|\mathbf{W}\|_1 + (\rho_t/2)\|\mathbf{W} - (\mathbf{Z}_{t+1} - (\mathbf{U}_t/\rho_t))\|_F^2$. According to [15], $\mathcal{S}_{\beta/\rho_t}(\mathbf{Z}_{t+1} - \frac{\mathbf{U}_t}{\rho_t})$ is the optimal solution for this problem. Similar to the algorithms in [10] and [21], we cannot theoretically prove that the algorithms for RKNNLRS and RKLRS converge to the global optimum, but they both converge well in practice as observed in our experiments.

VI. Experiments

In this section, we experimentally evaluate the efficiency and the effectiveness of our proposed approaches. In Section VI-A, we compare the efficiency of our proposed algorithm with that of the existing algorithms for solving the original LRR problem. In Section VI-B, we compare the effectiveness of our proposed RKLRR, RKLRS, and RKNNLRS algorithms with that of several subspace clustering methods for two real-world applications (i.e., face clustering and human activity clustering).

A. Efficiency Comparison for Different LRR Solvers

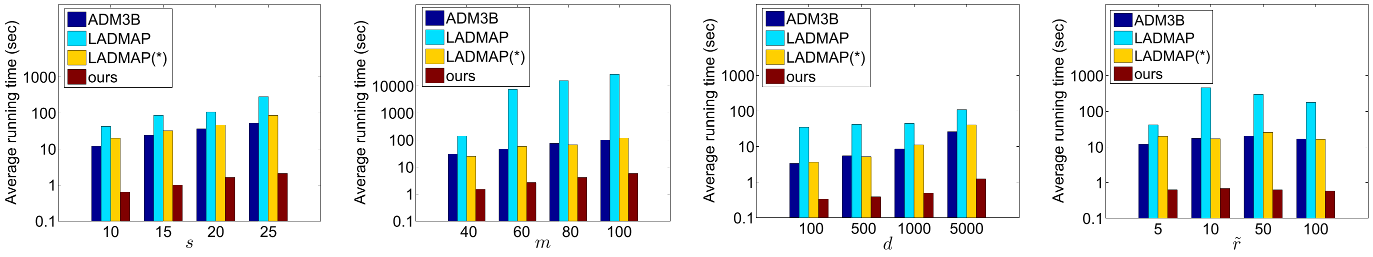
In this experiment, we compare our proposed algorithm (i.e., Algorithm 1 with the linear kernel) with the two existing LRR solvers ADM3B and LADMAP, for solving the original LRR problem on a synthetic data set. To reduce the computational cost, rank prediction is used in LADMAP when solving its nuclear-norm-related subproblem. However, since the subproblem may be solved inexactly, it may not achieve the global optimum. Therefore, we also implement a modified version of LADMAP [refered to as LADMAP(*)], in which we solve the nuclear-norm-related subproblem based on full SVD. For both ADM3B and LADMAP, we use the codes obtained from the homepages of the authors. For LADMAP, its fast version [18] is used. The experiments are performed on a computer with an Intel Xeon CPU (3.07 GHz) and 64-GB memory.

Following [1] and [18], we generate the synthetic data sets parameterized by $(s, m, d, \tilde{r})$. In particular, we construct s-independent subspaces with the bases $\{\mathbf{B}_i\}_{i=1}^s$ as follows: $\mathbf{B}_1$ is a $d \times \tilde{r}$ random orthogonal matrix, while $\{\mathbf{B}_i\}_{i=2}^s$ is generated by $\mathbf{B}_{i+1} = \mathbf{T}\mathbf{B}_i$, where $\mathbf{T}$ is a random rotation matrix. Accordingly, $\tilde{r}$ is the rank of each subspace, and d is the

TABLE I

RUNNING TIME, THE TOTAL NUMBERS OF ITERATIONS, AND THE OBJECTIVE VALUES OF FOUR ALGORITHMS FOR SOLVING THE ORIGINAL LRR PROBLEM ON THE SYNTHETIC DATA SET. $(s, m, d, \tilde{r})$ ARE SET TO (10, 20, 2000, 5). # ITERATIONS DENOTES THE TOTAL NUMBER OF ITERATIONS

	λ	0.01	0.03	0.05	0.07	0.1	0.3	0.5
RUNNING TIME	ADM3B	14.02	13.76	13.73	13.61	13.62	8.26	6.25
	LADMAP	0.89	1.69	1.70	6.75	15.50	130.97	135.48
	LADMAP(*)	1.00	1.57	4.60	17.61	27.44	41.83	44.67
	OURS	0.47	0.56	0.71	0.85	1.14	0.44	0.25
# ITERATIONS	ADM3B	319	314	313	311	309	198	147
	LADMAP	12	24	23	53	83	1000	1000
	LADMAP(*)	12	24	23	53	83	573	711
	OURS	10	17	22	25	27	11	3
OBJECTIVE VALUE	ADM3B	7.75	23.50	38.39	51.89	67.32	89.95	90.00
	LADMAP	7.07	21.20	35.34	49.21	66.16	2.27×10^{10}	4.65×10^{10}
	LADMAP(*)	7.07	21.20	35.34	49.21	66.16	89.97	90.02
	OURS	7.07	21.20	35.34	49.22	66.17	89.95	90.00



(a) (b) (c) (d)

Fig. 2. Average running time (in seconds) of each method for solving the original LRR problem on synthetic data sets generated using different values of parameters $(s, m, d, \tilde{r})$. This figure is best viewed in color. (a) Average running time versus s. (b) Average running time versus m. (c) Average running time versus d. (d) Average running time versus $\tilde{r}$.

ambient dimension of the data. Then, we sample m data points from each subspace using $\mathbf{X}_i = \mathbf{B}_i\mathbf{Q}_i$, $1 \le i \le s$, where $\mathbf{Q}_i \in \mathbb{R}^{\tilde{r}\times m}$ is a Gaussian matrix $\mathcal{N}(0, 1)$ with zero mean and unit variance. After that, 20% of the samples are randomly chosen to be corrupted. For each chosen sample $\mathbf{x}$, we add the Gaussian noise with zero mean and standard deviation $0.1\|\mathbf{x}\|$ on it. The running time, the total numbers of iterations, and the objective values of the four algorithms are reported in Table I. In addition, for a more comprehensive comparison of the average running time of these four algorithms, we vary s, m, d, and $\tilde{r}$, respectively, to generate synthetic data sets. In particular, the parameters $(s, m, d, \tilde{r})$ are first set to the default values (10, 20, 2000, 5). Then, each time, we only vary one parameter and fix the other parameters as the default values to generate several synthetic data sets. Accordingly, in Fig. 2(a)–(d), the corresponding average running time of each method, over all values of λ in Table I, is reported.

According to Table I, when setting λ to different values, our Algorithm 1 consistently converges with the minimum running time and the smallest number of iterations. Moreover, LADMAP sometimes does not converge to the global optimum, possibly because its nuclear-norm-related subproblem is not exactly solved when the rank prediction is inaccurate. However, its modified version LADMAP(*) and the proposed algorithm usually achieve almost the same objective values, which indicates that both of them converge well. For ADM3B, sometimes its objective value is obviously larger than the ones from Algorithm 1 and LADMAP(*), which indicates that ADM3B may not achieve a globally optimal solution in some cases. Moreover, as shown in Fig. 2, our algorithm is more efficient than others.

B. Real-World Applications

In this section, we compare the proposed methods with several baselines for two real-world applications: 1) face clustering and 2) human activity clustering.

1) Descriptions of the Data Sets: We use the following four publicly available data sets, including three face data sets and one human activity data set, to evaluate the performance of all the methods.

1) The **AR** data set [32] contains >4000 frontal face images from 126 individuals, where the images are with the variations in facial expressions, illumination, and occlusions (sun glasses and scarf). Following [13], we use the subset with 2600 images corresponding to 100 individuals (50 men and 50 women). We extract the 7080-D local binary patterns (LBPs) [33] to represent each face. In particular, after dividing each image (with the size of 120 × 165 pixels) into 8 × 15 nonoverlapping blocks, we extract a histogram of 59 bins from each block. We finally concatenate all these histograms to form a 7080-D feature vector for each image.

TABLE II

DETAILS OF FOUR DATA SETS FOR FACE CLUSTERING AND HUMAN ACTIVITY CLUSTERING. FOR SIMPLICITY, # SAMPLES (RESP., # CLUSTERS) DENOTES THE TOTAL NUMBER OF SAMPLES (RESP., CLUSTERS)

APPLICATION	DATASET	# SAMPLES	FEATURE DIMENSION	# CLUSTERS
FACE CLUSTERING	AR	2,600	7,080	100
	NH	4,660	1,937	5
	LFW	1,560	127,440	12
HUMAN ACTIVITY CLUSTERING	HARUS	2,947	561	6

Fig. 3. Top row: example face images from the AR data set. Middle row: example face images from the NH data set. Bottom row: example face images from the LFW data set.

2) The Notting Hill (NH) data set [34], [35] contains 4660 face images from the movie "Notting Hill," which correspond to five main casts. Since the face images are captured in the unconstrained environments, there are large variations in poses, facial expressions, illumination, and occlusions (see Fig. 3). Following [36], we extract the same type of descriptor, since it has shown good performance for representing faces in the wild. In particular, for each face image, 13 facial interesting points are detected. For each interest point, we extract the descriptor using the gray-level intensity values of pixels in the elliptical region centered at the interest point, which is further normalized to achieve local photometric invariance. Finally, the 13 descriptors are concatenated to form a 1937-D feature vector.
3) The Labeled Face in the Wild (LFW) data set [37] contains more than 13,000 face images collected in an unconstrained environment. Similar to the ones in the NH data set, the faces in the LFW data set are also with large variations in poses, facial expressions, illumination, and occlusions, thus making the face clustering task very challenging. Considering that many individuals in this data set only have few photos that are insufficient for the clustering task, we use the subset containing the subjects with more than 50 photos, which consists of 1560 face images from 12 individuals. To represent each face, we use the 127,440-D LBP feature provided by [38] because of its excellent performance on this challenging data set. Please refer to [38] for the feature extraction details.
4) The Human Activity Recognition Using Smartphones (HARUSs) data set [39] contains the data collected using embedded sensors (i.e. accelerometer and gyroscope) on the smartphones. In particular, the smartphones with embedded sensors are carried by volunteers on their waists, while they are conducting daily activities (e.g., walking, sitting, and laying). The captured sensor signals (three-axial linear acceleration and three-axial angular velocity) are preprocessed to filter the noise and postprocessed (e.g., sampling). Finally, for each signal, a 561-D feature vector with time and frequency domain variables is extracted. We use the testing set containing 2947 signals related to six activities for a performance evaluation.

Details of these real-world data sets are summarized in Table II. Example faces from the face data sets are shown in Fig. 3.

2) Baselines and Evaluation Criterion: We compare our proposed kernel-based methods (i.e., RKLRR, RKLRS, and RKNNLRS) with the corresponding counterparts (i.e., LRR, LRS, and NNLRS). Moreover, the following state-of-the-art subspace clustering methods are also compared as baselines: 1) local subspace affinity (LSA) [9]; 2) spectral curvature clustering (SCC) [40]; 3) low-rank subspace clustering (LRSC) [11] using the Frobenius norm [LRSC(F)] and the ℓ_1 norm [LRSC(1)]; 4) SSC [10]; and 5) the kernel-based methods—kernel SSC [41], kernel SCC (KSCC) [42], and SSDL [26].

In particular, for all the kernel-based methods, we adopt the histogram intersection kernel (HIK) [43] on the AR and LFW data sets, because HIK has shown good performance when dealing with frequency-based features such as LBP. Moreover, we adopt the commonly used Gaussian kernel on the NH and HARUS data sets. For all the LRR-based methods, we perform subspace clustering as suggested in [21]. In particular, we first compute the affinity matrix based on the solution $\mathbf{Z}$, and then apply spectral clustering (e.g., NCut [12]) on the resultant affinity matrix. Considering LSA, SSC, LRSC(F) and LRSC(1) cannot directly handle the high-dimensional feature on the LFW dataset, we reduce the feature dimension before applying these methods on this dataset, in which principal component analysis (PCA) is used to preserve 90% of the energy. For LRR [21], the convergence issue of the original optimization algorithm in [21] may degrade the clustering performance, so we use Algorithm 1 in this paper with the linear kernel ($\mathbf{K} = \mathbf{X}'\mathbf{X}$) when reporting the results of LRR in Table III.

We use the *clustering error* [10] for performance evaluation, which is defined as

$$\textit{clustering error} = \frac{\text{\# misclassified samples}}{\text{\# all samples}}$$

where # denotes "the number of." Lower clustering error indicates better performance. Following [1], to fairly compare all the methods, we manually tune the parameters (excluding the kernel parameters) of all methods and report the best

TABLE III

CLUSTERING ERRORS (%) OF ALL METHODS ON THE AR, NH, AND LFW DATA SETS (FOR FACE CLUSTERING) AND THE HARUS DATA SET (FOR HUMAN ACTIVITY CLUSTERING). THE METHODS WITH THE SUPERSCRIPT * ARE OUR PROPOSED ONES. THE BEST RESULTS ARE IN BOLD

DATASET	LSA	LRSC(1)	LRSC(F)	SSC	KSSC	SCC	KSCC	SSDL	LRR	LRS	NNLRS	RKLRR*	RKLRS*	RKNNLRS*
AR	26.42	9.50	9.38	10.65	6.77	92.89	92.42	5.58	6.96	5.69	10.36	4.69	**4.35**	8.86
NH	13.78	18.28	14.85	14.25	9.16	45.17	22.73	5.19	15.45	10.15	10.43	5.06	**4.33**	7.94
LFW	27.31	39.42	37.69	13.97	13.40	71.80	73.53	13.97	12.37	10.13	10.38	12.31	8.59	**5.71**
HARUS	20.09	28.40	27.93	32.13	24.60	32.92	29.25	18.90	32.98	27.31	26.50	18.19	**18.15**	24.40

performance for each method. Note that the HIK is parameter-free. For Gaussian kernel, we use the default bandwidth parameter, which is set to the mean of the distances between all the samples.

3) Experimental Results: The clustering errors for all methods on the AR, NH, LFW and HARUS datasets are shown in Table III. According to Table III, we have the following observations.

1) Our kernel-based methods RKLRR, RKLRS, and RKNNLRS perform better than the corresponding baselines LRR, LRS, and NNLRS, respectively. The improvements brought by kernelization are generally more than 2% on these data sets. RKLRS achieves the best results on three data sets (AR, NH, and HARUS), while RKNNLRS achieves the best result on the LFW data set. SCC and KSCC do not explicitly handle corrupted data, so their clustering errors are large, especially on the AR data set, where some faces are occluded by sunglasses and scarves. Such an observation is consistent with that in [10].
2) LRS and NNLRS (both are the variants of LRR) perform better than LRR. Regarding the corresponding kernelized versions, RKLRS consistently outperforms RKLRR. Interestingly, although RKNNLRS has the additional nonnegative constraint, it generally performs worse than RKLRS.
3) On all four data sets, our proposed RKLRR consistently achieves better performance than that of SSDL, which uses the squared Frobenius norm to regularize the representation error. This observation demonstrates the robustness of the $\ell_{2,1}$ norm in RKLRR to handle the outliers.
4) When the feature dimension is very high, several approaches (e.g., LSA) that work in the linear feature space may need to first preprocess the data by reducing the dimension with the existing dimension reduction techniques (such as PCA used in our experiments). In contrast, the kernel-based methods still work well, which demonstrates the advantage of the kernel-based methods when dealing with high-dimensional data.

Moreover, we take the NH data set (where $d < n$) and the LFW data set (where $d > n$) as two representative examples to compare the efficiency of Algorithm 1 with the existing optimization algorithms ADM3B and LADMAP for solving the original LRR problem. Using the same computer with the configuration mentioned in Section VI-A, the recorded running times (in seconds) are 5970.52, 2759.86, and 879.84 (resp., 10 114.25, 609.02, and 81.39) for ADM3B, LADMAP, and Algorithm 1 on the NH (resp., LFW) data set, demonstrating the efficiency of our optimization algorithm especially when the feature dimension is high.

VII. CONCLUSION

In this paper, we have proposed the kernelized version of LRR for handling clean data, and presented a closed-form solution for it. Moreover, to handle the corrupted data, we have proposed RKLRR as well as a new optimization algorithm to solve the corresponding nontrivial optimization problem. In particular, we provide the closed-form solution for the $\ell_{2,1}$ norm-related subproblem, such that both subproblems involved in our optimization algorithm can be efficiently solved. Moreover, the convergence of our algorithm is guaranteed in theory. With our proposed optimization technique, many variants of LRR with the $\ell_{2,1}$ norm-based regularizer on the error term can be similarly kernelized and solved, for which we take NNLRS and its simplified version LRS as two showcases to introduce their kernelized versions. Comprehensive experiments on the synthetic data sets and the real-world data sets have clearly demonstrated the efficiency of our Algorithm 1 and the effectiveness of RKLRR as well as the kernelized versions of NNLRS and LRS.

In the future, we plan to study how to choose the optimal kernels for our RKLRR, RKLRS, and RKNNLRS.

APPENDIX A
PROOF OF THEOREM 1

Proof: Since $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]$, we can decompose the equality constraint of problem (8) as

$$\phi(\mathbf{x}_j) = \sum_{i=1}^{n} \phi(\mathbf{x}_i) Z_{ij} \quad \forall j = 1, \ldots, n. \tag{45}$$

Note that each equality in (45) can be rewritten as

$$0 = \sum_{i=1}^{n} \phi(\mathbf{x}_i)(Z_{ij} - \delta_{ij}) \tag{46}$$

where the scalar $\delta_{ij} \in \{0, 1\}$ is the (i, j)th element of $\mathbf{I}_n$, $\forall i, j = 1, \ldots, n$. Note that $\sum_{i=1}^{n} \phi(\mathbf{x}_i)(Z_{ij} - \delta_{ij})$ is zero, if and only if the inner product between $\sum_{i=1}^{n} \phi(\mathbf{x}_i)(Z_{ij} - \delta_{ij})$ and itself is zero, namely

$$\left(\sum_{i=1}^{n} \phi(\mathbf{x}_i)(Z_{ij} - \delta_{ij})\right)' \left(\sum_{i=1}^{n} \phi(\mathbf{x}_i)(Z_{ij} - \delta_{ij})\right) = 0. \tag{47}$$

By defining $\mathbf{z}_j$ (resp., $\mathbf{e}_j$) as the jth column of $\mathbf{Z}$ (resp., $\mathbf{I}_n$) and considering $\mathbf{K} = \mathbf{V}\Sigma^2\mathbf{V}'$, we rewrite the left-hand side of (47) as

$$\begin{aligned}&(\Phi(\mathbf{X})(\mathbf{z}_j - \mathbf{e}_j))'(\Phi(\mathbf{X})(\mathbf{z}_j - \mathbf{e}_j))\\ &\quad= (\mathbf{z}_j - \mathbf{e}_j)'\Phi(\mathbf{X})'\Phi(\mathbf{X})(\mathbf{z}_j - \mathbf{e}_j)\\ &\quad= (\mathbf{z}_j - \mathbf{e}_j)'\mathbf{K}(\mathbf{z}_j - \mathbf{e}_j)\\ &\quad= (\mathbf{z}_j - \mathbf{e}_j)'\mathbf{V}\Sigma^2\mathbf{V}'(\mathbf{z}_j - \mathbf{e}_j)\\ &\quad= (\mathbf{V}'\mathbf{z}_j - \mathbf{V}'\mathbf{e}_j)'\Sigma^2(\mathbf{V}'\mathbf{z}_j - \mathbf{V}'\mathbf{e}_j).\end{aligned}$$

In summary, the probelm in (8) can be rewritten as follows:

$$\begin{aligned}&\min_{\mathbf{Z}=[\mathbf{z}_1,\ldots,\mathbf{z}_n]} \quad \|\mathbf{Z}\|_*\\ &\text{s.t. } (\mathbf{V}'\mathbf{z}_j - \mathbf{V}'\mathbf{e}_j)'\Sigma^2(\mathbf{V}'\mathbf{z}_j - \mathbf{V}'\mathbf{e}_j) = 0 \quad \forall j = 1,\ldots,n.\end{aligned} \tag{48}$$

Note that $\|\mathbf{Z}\|_* = \|\mathbf{V}\mathbf{Z}\|_*$ always holds, since $\mathbf{V}$ is an orthogonal matrix and the nulcear norm is unitarily invariant. Let us define $\tilde{\mathbf{Z}} = \mathbf{V}'\mathbf{Z}$, and let $\tilde{\mathbf{z}}_j = \mathbf{V}'\mathbf{z}_j$ denote the jth column of $\tilde{\mathbf{Z}}$. As a result, problem (48) becomes

$$\begin{aligned}&\min_{\tilde{\mathbf{Z}}=[\tilde{\mathbf{z}}_1,\ldots,\tilde{\mathbf{z}}_n]} \quad \|\tilde{\mathbf{Z}}\|_*\\ &\text{s.t. } (\tilde{\mathbf{z}}_j - \mathbf{V}'\mathbf{e}_j)'\Sigma^2(\tilde{\mathbf{z}}_j - \mathbf{V}'\mathbf{e}_j) = 0 \quad \forall j = 1,\ldots,n.\end{aligned} \tag{49}$$

Since $\Sigma^2 = \text{diag}([\sigma_1^2,\ldots,\sigma_{r_K}^2, 0,\ldots,0]')$, where $\{\sigma_i\}_{i=1}^{r_K}$ are positive, the equality $(\tilde{\mathbf{z}}_j - \mathbf{V}'\mathbf{e}_j)'\Sigma^2(\tilde{\mathbf{z}}_j - \mathbf{V}'\mathbf{e}_j) = 0$ holds if and only if the first r_K elements of the vector $(\tilde{\mathbf{z}}_j - \mathbf{V}'\mathbf{e}_j)$ are zeros, $\forall j = 1,\ldots,n$. Recalling that $\mathbf{V}_K$ is obtained using the first r_K columns of $\mathbf{V}$, we can equivalently rewrite the constraint of (49) as $\tilde{\mathbf{Z}} \in \mathcal{Z}$, where the set $\mathcal{Z}$ is defined as

$$\mathcal{Z} = \{\tilde{\mathbf{Z}} \mid \tilde{\mathbf{Z}} = [\mathbf{V}_K, \mathbf{R}]',\ \mathbf{R} \in \mathbb{R}^{n\times(n-r_K)}\}.$$

Accordingly, the problem in (49) becomes

$$\min_{\tilde{\mathbf{Z}}} \|\tilde{\mathbf{Z}}\|_* \quad \text{s.t.} \quad \tilde{\mathbf{Z}} \in \mathcal{Z}. \tag{50}$$

Note that $\|[\mathbf{A},\mathbf{B}]'\|_* \geq \|\mathbf{A}\|_*$ holds for any matrices $\mathbf{A}$ and $\mathbf{B}$ with compatible sizes, and the inequality becomes an equality if and only when $\mathbf{B}$ is a zero matrix. Therefore, for any $\tilde{\mathbf{Z}} = [\mathbf{V}_K, \mathbf{R}]$, where $\mathbf{R} \in \mathbb{R}^{n\times(n-r_K)}$, we have

$$\|\tilde{\mathbf{Z}}\|_* = \|[\mathbf{V}_K, \mathbf{R}]'\|_* \geq \|\mathbf{V}_K'\|_* = \|[\mathbf{V}_K, \mathbf{O}_{n\times(n-r_K)}]'\|_*.$$

As a result, $[\mathbf{V}_K, \mathbf{O}_{n\times(n-r_K)}]'$ is the optimal solution of the problem in (50). Considering $\mathbf{Z} = \mathbf{V}\tilde{\mathbf{Z}}$ (due to $\tilde{\mathbf{Z}} = \mathbf{V}'\mathbf{Z}$), we conclude that $\mathbf{V}[\mathbf{V}_K, \mathbf{O}_{n\times(n-r_K)}]'$, which can also be rewritten as $\mathbf{V}_K\mathbf{V}_K'$, is the optimal solution of (8). □

APPENDIX B
PROOF OF LEMMA 1

Proof: For convenience, we denote the objective of (18) as

$$F(\mathbf{p}) = f(\mathbf{p}) + \psi(\mathbf{p})$$

where $f(\mathbf{p}) = (\mathbf{p}'\mathbf{S}^2\mathbf{p})^{1/2}$ and $\psi(\mathbf{p}) = \tau/2\|\mathbf{p} - \mathbf{h}\|^2$.

Note that $F(\mathbf{p})$ is convex, and an optimum of (18) corresponds to a stationary point of $F(\mathbf{p})$. Therefore, to find the optimal solution of the problem in (18), we first derive the subgradient of $F(\mathbf{p})$, and then seek its stationary point.

Since $F(\mathbf{p})$ can be expressed as the sum of $f(\mathbf{p})$ and $\psi(\mathbf{p})$, let us derive the subgradients of $f(\mathbf{p})$ and $\psi(\mathbf{p})$, respectively. The subgradient of $\psi(\mathbf{p})$ with respect to $\mathbf{p}$ is as follows:

$$\partial\psi(\mathbf{p}) = \tau(\mathbf{p} - \mathbf{h}).$$

For $f(\mathbf{p})$, note that it can be rewritten as $f(\mathbf{p}) = \|\mathbf{S}\mathbf{p}\|$. Based on [44], the subgradient of $f(\mathbf{p})$ with respect to $\mathbf{p}$ is as follows:

$$\partial f(\mathbf{p}) = \begin{cases}\{\mathbf{S}'\mathbf{r} \mid \|\mathbf{r}\| \leq 1\}, & \text{if } \mathbf{S}\mathbf{p} = \mathbf{0}_q\\ \frac{\mathbf{S}^2\mathbf{p}}{\|\mathbf{S}\mathbf{p}\|}, & \text{otherwise.}\end{cases}$$

Recall that the diagonal matrix $\mathbf{S}$, in which all the diagonal elements are positive, is a positive definite matrix. As a result, $\mathbf{S}\mathbf{p} = \mathbf{0}_q$ is equivalent to $\mathbf{p} = \mathbf{0}_q$.

In summary, the subgradient of $F(\mathbf{p})$ can be expressed as

$$\partial F(\mathbf{p}) = \begin{cases}\{\mathbf{S}'\mathbf{r} + \tau(\mathbf{p} - \mathbf{h}) \mid \|\mathbf{r}\| \leq 1\}, & \text{if } \mathbf{p} = \mathbf{0}_q\\ \dfrac{\mathbf{S}^2\mathbf{p}}{\|\mathbf{S}\mathbf{p}\|} + \tau(\mathbf{p} - \mathbf{h}), & \text{otherwise.}\end{cases}$$

Based on the above two cases, let us discuss the stationary point of $F(\mathbf{p})$ accordingly in two cases as follows.

1) When $\mathbf{p} = \mathbf{0}_q$, we have

$$\partial F(\mathbf{p}) = \{\mathbf{S}'\mathbf{r} + \tau(\mathbf{p} - \mathbf{h}) \mid \|\mathbf{r}\| \leq 1\}.$$

Note that $\mathbf{p}^* = \mathbf{0}_q$ is a stationary point, if and only if

$$\mathbf{0}_q \in \{\mathbf{S}'\mathbf{r} + \tau(\mathbf{p}^* - \mathbf{h}) \mid \|\mathbf{r}\| \leq 1\}. \tag{51}$$

In other words, $\mathbf{p}^* = \mathbf{0}_q$ is a stationary point, if and only if there is a vector $\mathbf{r}$ that satisfies the following two conditions:

$$\mathbf{S}'\mathbf{r} + \tau(\mathbf{p}^* - \mathbf{h}) = \mathbf{0}_q \tag{52}$$

$$\|\mathbf{r}\| \leq 1. \tag{53}$$

Recalling that $\mathbf{p}^* = \mathbf{0}_q$ and $\mathbf{S}$ is positive definite, (52) is equivalent to $\mathbf{r} = \tau\mathbf{S}^{-1}\mathbf{h}$. Combining this with inequality (53), we arrive at

$$\|\mathbf{S}^{-1}\mathbf{h}\| \leq \frac{1}{\tau}.$$

Therefore, $\mathbf{p}^* = \mathbf{0}_q$ is a stationary point of $F(\mathbf{p})$, if and only if $\|\mathbf{S}^{-1}\mathbf{h}\| \leq (1/\tau)$. In particular, when $\|\mathbf{S}^{-1}\mathbf{h}\| \leq (1/\tau)$, we have $\mathbf{r} = \tau\mathbf{S}^{-1}\mathbf{h}$ that satisfies (51).

2) When $\mathbf{p} \neq \mathbf{0}_q$, we have

$$\partial F(\mathbf{p}) = \frac{\mathbf{S}^2\mathbf{p}}{\|\mathbf{S}\mathbf{p}\|} + \tau(\mathbf{p} - \mathbf{h}).$$

As a result, $\mathbf{p}^* \neq \mathbf{0}_q$ is a stationary point, if and only if $\mathbf{S}^2\mathbf{p}^*/\|\mathbf{S}\mathbf{p}^*\| + \tau(\mathbf{p}^* - \mathbf{h}) = \mathbf{0}_q$. This condition can be rewritten as

$$\left(\frac{\mathbf{S}^2}{\|\mathbf{S}\mathbf{p}^*\|} + \tau\mathbf{I}\right)\mathbf{p}^* = \tau\mathbf{h}.$$

By defining a scalar $\alpha \triangleq \|\mathbf{S}\mathbf{p}^*\|$, we rewrite $\mathbf{p}^*$ as

$$\mathbf{p}^* = \left(\frac{\mathbf{S}^2}{\tau\alpha} + \mathbf{I}\right)^{-1}\mathbf{h}. \tag{54}$$

Since $\mathbf{p}^* \neq \mathbf{0}_q$, we have $\mathbf{Sp}^* \neq \mathbf{0}_q$. Accordingly, we have $\alpha = \|\mathbf{Sp}^*\| > 0$. Moreover, given $\alpha = \|\mathbf{Sp}^*\|$ and $\alpha > 0$, we conclude that α is the positive root of

$$\alpha^2 = (\mathbf{p}^*)'\mathbf{S}^2\mathbf{p}^*. \tag{55}$$

By substituting (54) into (55), we obtain

$$\alpha^2 = \tau^2\alpha^2\mathbf{h}'\mathrm{diag}\left(\left\{\frac{s_i^2}{(s_i^2+\tau\alpha)^2}\right\}_{1\le i\le q}\right)\mathbf{h}. \tag{56}$$

Dividing both the sides of (56) by $\tau^2\alpha^2$, we arrive at (20).

Now, if the positive root of the equation in (20) exists, we can first obtain α, and then obtain $\mathbf{p}^*$ by substituting α into (54). In the following, we show the existence condition of the positive root of (20).

For convenience, we define the following function with respect to α:

$$\ell(\alpha) \triangleq \mathbf{h}'\mathrm{diag}\left(\left\{\frac{s_i^2}{(\tau\alpha+s_i^2)^2}\right\}_{1\le i\le q}\right)\mathbf{h} - \frac{1}{\tau^2}. \tag{57}$$

With this definition, (20) can be rewritten as $\ell(\alpha) = 0$. We can verify the following properties of $\ell(\alpha)$.

1) The function $\ell(\alpha)$ with respect to α is continuous, and it is strictly decreasing when α is in the range $[0, +\infty)$. In fact, the continuity of $\ell(\alpha)$ is obvious. Moreover, we can easily verify that the gradient of $\ell(\alpha)$ is negative for any positive α, so $\ell(\alpha)$ is strictly decreasing when α is in the range $[0, +\infty)$.
2) $\lim_{\alpha\to 0}\ell(\alpha) = \|\mathbf{S}^{-1}\mathbf{h}\|^2 - 1/\tau^2$.
Actually, due to the continuity of $\ell(\alpha)$, $\lim_{\alpha\to 0}\ell(\alpha)$ can be calculated as $\ell(0) = \mathbf{h}'\mathrm{diag}(\{s_i^{-2}\}_{1\le i\le q})\mathbf{h} - 1/\tau^2$, where the first term can be expressed as $\|\mathbf{S}^{-1}\mathbf{h}\|^2$.
3) $\lim_{\alpha\to+\infty}\ell(\alpha) < 0$.
Note that $\lim_{\alpha\to+\infty}\ell(\alpha) = -1/\tau^2$ and $\tau^2 > 0$, so $\lim_{\alpha\to+\infty}\ell(\alpha) < 0$.

Based on the above properties and the intermediate value theorem [31], it is easy to verify that there exists a positive scalar α which satisfies $\ell(\alpha) = 0$ if and only if $\|\mathbf{S}^{-1}\mathbf{h}\|^2 - (1/\tau^2) > 0$, namely

$$\|\mathbf{S}^{-1}\mathbf{h}\| > \frac{1}{\tau}.$$

In particular, when $\|\mathbf{S}^{-1}\mathbf{h}\| > 1/\tau$, the positive root of the equation $\ell(\alpha) = 0$ exists, and such positive root is unique due to the strictly decreasing property of $\ell(\alpha)$. Furthermore, let α^* be the unique positive root of $\ell(\alpha) = 0$, then we can prove the following inequality that determines the range of α^*:

$$\max(0, \alpha_l) \le \alpha^* \le \alpha_u \tag{58}$$

in which $\alpha_u = (\mathbf{h}'\mathrm{diag}(\{s_i^2\}_{1\le i\le q})\mathbf{h})^{1/2} - s_q^2/\tau$ is the positive root of an equation $f_u(\alpha) = 0$; $\alpha_l = (\mathbf{h}'\mathrm{diag}(\{s_i^2\}_{1\le i\le q})\mathbf{h})^{1/2} - s_1^2/\tau$ is the larger root of another equation $f_l(\alpha) = 0$, where the two functions $f_u(\alpha)$ and $f_l(\alpha)$ are defined as

$$f_u(\alpha) = \mathbf{h}'\frac{\mathrm{diag}(\{s_i^2\}_{1\le i\le q})}{(\tau\alpha+s_q^2)^2}\mathbf{h} - \frac{1}{\tau^2}$$

$$f_l(\alpha) = \mathbf{h}'\frac{\mathrm{diag}(\{s_i^2\}_{1\le i\le q})}{(\tau\alpha+s_1^2)^2}\mathbf{h} - \frac{1}{\tau^2}.$$

In particular, $f_u(\alpha)$ is obtained by the amplification of $\ell(\alpha)$ by replacing $(\tau\alpha+s_i^2)$ with $(\tau\alpha+s_q^2)$, while $f_l(\alpha)$ is obtained by the minification of $\ell(\alpha)$ by replacing $(\tau\alpha+s_i^2)$ with $(\tau\alpha+s_1^2)$.

In fact, the following statements can be easily verified.

1) When $\alpha \in [0, +\infty)$, $f_u(\alpha)$ and $f_l(\alpha)$ are continuous, strictly decreasing with respect to α, and satisfy
$$f_u(\alpha) \ge \ell(\alpha) \ge f_l(\alpha).$$
2) $\lim_{\alpha\to 0} f_u(\alpha) = f_u(0) \ge \ell(0) > 0$ holds, while it is unclear whether $\lim_{\alpha\to 0} f_l(\alpha) = f_l(0)$ is positive or not.
3) $\lim_{\alpha\to+\infty} f_u(\alpha) = \lim_{\alpha\to+\infty} f_l(\alpha) = -1/\tau^2 < 0$.

Therefore, $f_u(\alpha) = 0$ has a unique positive root, namely α_u, which is no less than α^*. Moreover, α_l [i.e., the larger root of $f_l(\alpha) = 0$] is no greater than α^*, but not necessarily positive. As a result, the inequalities in (58) are verified. Accordingly, we can obtain α^* by the bisection search method [31] within the searching range $[\max(0, \alpha_l), \alpha_u]$.

In summary, if and only if $\|\mathbf{S}^{-1}\mathbf{h}\| > (1/\tau)$ is satisfied, there exists $\mathbf{p}^* \neq \mathbf{0}_q$, which is optimal to the problem in (18). In particular, when $\|\mathbf{S}^{-1}\mathbf{h}\| > (1/\tau)$, $\mathbf{p}^*$ can be calculated as in (54), where α is the unique positive root of the equation in (20) and can be obtained by the bisection search method [31].

This completes the proof of Lemma 1. □

Acknowledgment

This research was supported by funding from the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

References

[1] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.

[2] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Int. J. Comput. Vis.*, vol. 100, no. 3, pp. 217–240, 2012.

[3] Y. Zhang, Z. Sun, R. He, and T. Tan, "Robust subspace clustering via half-quadratic minimization," in *Proc. 14th IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 3096–3103.

[4] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 430–437.

[5] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 3827–3833.

[6] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. 16th IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Madison, WI, USA, Jun. 2003, pp. I-11–I-18.

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

[8] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[9] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Proc. 9th Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 94–106.

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

[11] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.

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

[13] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2008.

[14] H. Fu, D. Xu, S. Lin, D. W. K. Wong, and J. Liu, "Automatic optic disc detection in OCT slices via low-rank reconstruction," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 4, pp. 1151–1158, Apr. 2015.

[15] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2328–2335.

[16] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge Univ. Press, 2004.

[17] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 749–761, May 2013.

[18] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. 24th Annu. Conf. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 612–620.

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

[20] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

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

[22] Z. Zeng *et al.*, "Learning by associating ambiguously labeled images," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 708–715.

[23] M. Tan, I. W. Tsang, L. Wang, B. Vandereycken, and S. J. Pan, "Riemannian pursuit for big matrix recovery," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 1539–1547.

[24] B. He, M. Tao, and X. Yuan, "Alternating direction method with Gaussian back substitution for separable convex programming," *SIAM J. Optim.*, vol. 22, no. 2, pp. 313–340, 2012.

[25] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[26] J. Wang, V. Saligrama, and D. A. Castañón, "Structural similarity and distance in learning," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 2011, pp. 744–751.

[27] S. Xiao, W. Li, D. Xu, and D. Tao, "FaLRR: A fast low rank representation solver," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 4612–4620.

[28] N. Cristianini, J. Shawe-Taylor, and J. S. Kandola, "Spectral kernel methods for clustering," in *Proc. 14th Annu. Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2001, pp. 649–655.

[29] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, 2008.

[30] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[31] R. L. Burden and J. D. Faires, *Numerical Analysis*. Boston, MA, USA: Cengage Learning, 2011.

[32] A. M. Martinez and R. Benavente, "The AR face database," CVC, Barcelona, Spain, Tech. Rep. 24, Jun. 1998.

[33] M. Kan, D. Xu, S. Shan, W. Li, and X. Chen, "Learning prototype hyperplanes for face verification in the wild," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3310–3316, Aug. 2013.

[34] Y.-F. Zhang, C. Xu, H. Lu, and Y.-M. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.

[35] S. Xiao, M. Tan, and D. Xu, "Weighted block-sparse low rank representation for face clustering in videos," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 123–138.

[36] M. Everingham, J. Sivic, and A. Zisserman, "'Hello! My name is... Buffy'—Automatic naming of characters in TV video," in *Proc. 17th Brit. Mach. Vis. Conf.*, Edinburgh, U.K., Sep. 2006, pp. 899–908.

[37] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[38] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3025–3032.

[39] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. 4th Int. Workshop Ambient Assist. Living Home Care*, Vitoria-Gasteiz, Spain, Dec. 2012, pp. 216–223.

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

[41] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. 21st IEEE Int. Conf. Image Process.*, Paris, France, Oct. 2014, pp. 2849–2853.

[42] G. Chen, S. Atev, and G. Lerman, "Kernel spectral curvature clustering (KSCC)," in *Proc. 12th IEEE Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, Sep./Oct. 2009, pp. 765–772.

[43] J. Wu and J. M. Rehg, "Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Sep./Oct. 2009, pp. 630–637.

[44] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 569–592, 2009.

Shijie Xiao (S'15) received the B.Eng. degree from the Honors School, Harbin Institute of Technology, Harbin, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore.

His current research interests include machine learning and computer vision.

Mingkui Tan (M'14) received the bachelor's degree in environmental science and engineering and the master's degree in control science and engineering from Hunan University, Changsha, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014.

He is currently a Senior Research Associate with the School of Computer Science, The University of Adelaide, Adelaide, SA, Australia. His current research interests include compressive sensing, big data learning, and large-scale optimization.

Dong Xu (M'07–SM'13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2005, respectively.

He was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Hong Kong, for over two years, while pursuing the Ph.D. degree. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, USA, for one year. He was a Faculty Member with the School of Computer Engineering, Nanyang Technological University, Singapore. He is currently a Faculty Member with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW, Australia. His current research interests include computer vision, statistical learning, and multimedia content analysis.

Dr. Xu has co-authored a paper that received the Best Student Paper Award in the IEEE International Conference on Computer Vision and Pattern Recognition in 2010. His co-authored work also won the IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award in 2014.

Zhao Yang Dong (M'99–SM'06) received the Ph.D. degree from The University of Sydney, Sydney, NSW, Australia, in 1999.

He was the Ausgrid Chair and Director of the Centre for Intelligent Electricity Networks with the University of Newcastle, Callaghan, NSW, Australia. He held academic and industrial positions with the Hong Kong Polytechnic University, Hong Kong, and TASNetworks, Lenah Valley, TAS, Australia. He is currently a Professor and the Head of the School of Electrical and Information Engineering with The University of Sydney. His current research interests include smart grid, power system planning, power system security, load modeling, renewable energy systems, electricity market, and computational intelligence and its application in power engineering.

Prof. Dong is an Editor of the IEEE TRANSACTIONS ON SMART GRID, the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, the IEEE POWER ENGINEERING LETTERS, and *IET Renewable Power Generation*.