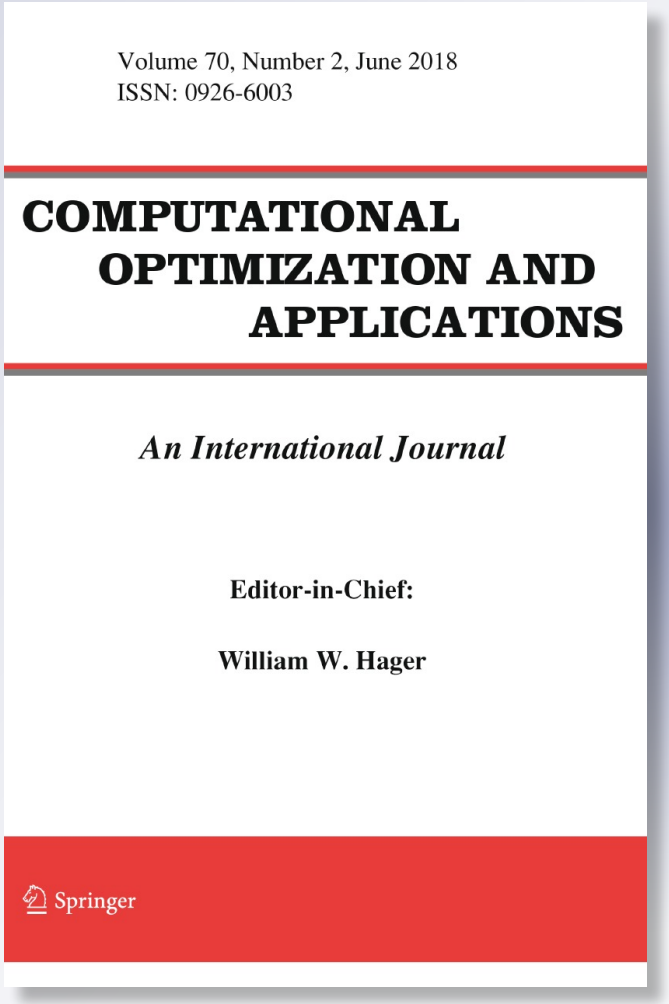
A nonconvex formulation for low rank subspace clustering: algorithms and convergence analysis

Hao Jiang, Daniel P. Robinson, René Vidal & Chong You

Computational Optimization and Applications
An International Journal

ISSN 0926-6003
Volume 70
Number 2

Comput Optim Appl (2018) 70:395-418
DOI 10.1007/s10589-018-0002-6



Springer

Comput Optim Appl (2018) 70:395–418
https://doi.org/10.1007/s10589-018-0002-6

A nonconvex formulation for low rank subspace clustering: algorithms and convergence analysis

Hao Jiang[1] · **Daniel P. Robinson**[1] · **René Vidal**[1] · **Chong You**[1]

Received: 14 July 2017 / Published online: 27 March 2018

Abstract We consider the problem of subspace clustering with data that is potentially corrupted by both dense noise and sparse gross errors. In particular, we study a recently proposed low rank subspace clustering approach based on a nonconvex modeling formulation. This formulation includes a nonconvex spectral function in the objective function that makes the optimization task challenging, e.g., it is unknown whether the alternating direction method of multipliers (ADMM) framework proposed to solve the nonconvex model formulation is provably convergent. In this paper, we establish that the spectral function is differentiable and give a formula for computing the derivative. Moreover, we show that the derivative of the spectral function is Lipschitz continuous and provide an explicit value for the Lipschitz constant. These facts are then used to provide a lower bound for how the penalty parameter in the ADMM method should be chosen. As long as the penalty parameter is chosen according to this bound, we show that the ADMM algorithm computes iterates that have a limit point satisfying first-order optimality conditions. We also present a second strategy for solving the nonconvex problem that is based on proximal gradient calculations. The convergence and performance of the algorithms is verified through experiments on real data from face and digit clustering and motion segmentation.

✉ Daniel P. Robinson
daniel.p.robinson@gmail.com

Hao Jiang
jianghaohku@gmail.com

René Vidal
rvidal@cis.jhu.edu

Chong You
cyou6@jhu.edu

[1] Johns Hopkins University, Baltimore, MD 21218, USA

Keywords ADMM · Nonconvex · Subspace clustering

1 Introduction

In many computer vision and pattern recognition applications such as motion segmentation [44], face clustering [20], and image processing [21], the high-dimensional data we observe lie approximately in a union of low-dimensional subspaces. The task of subspace clustering is to automatically identify the number of subspaces, the dimension of each subspace, and the data membership, i.e., to partition the data by assigning each data point to its corresponding low-dimensional subspace. See Fig. 1 for examples of data for face clustering and motion segmentation.

Formally, let $X \in \mathbb{R}^{D\times N}$ denote a data matrix where each column is a data point in $\mathbb{R}^D$ obtained by adding noise and sparse errors to a clean data point drawn from among a collection of unknown subspaces of unknown dimensions. Subspace clustering seeks to cluster the data points from X into groups such that the points generated from the same subspace are in the same group. This is an unsupervised learning problem since the subspaces and their dimensions are all assumed to be unknown and therefore must be learned from the data matrix X only.

1.1 Prior work on subspace clustering

Many methods have been developed for subspace clustering that include algebraic, iterative, statistical, and spectral clustering based methods (see [41,42] and the references therein). Overall, the best performance has been achieved by spectral clustering methods [42], which proceed in two stages. In the first stage, they compute an affinity matrix from the data that encodes the similarity between pairs of data points. During the second stage, a weighted graph is constructed from the affinity matrix and spectral clustering [45] on the graph is performed. The accuracy of the spectral clustering stage depends critically on the computation of an appropriate affinity matrix in stage one. For this reason an extensive amount of research has focused on computing excellent affinity matrices, which is also the focus of this paper.

Many recent spectral clustering based methods compute the affinity matrix by capitalizing on the *self-expressiveness property* that was first proposed in [14]. The data

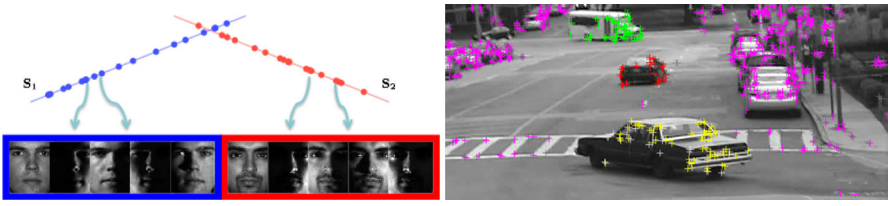


Fig. 1 Left figure: face images from multiple individuals under different lighting conditions can be well approximated by multiple low-dimensional subspaces denoted by S_1 and S_2. Right figure: marker trajectories from multiple rigid objectives can be well approximated by multiple low-dimensional subspaces

matrix X is said to satisfy the self-expressiveness property if there exists $C \in \mathbb{R}^{N\times N}$ such that $X = XC$ and $\text{diag}(C) = 0$, where $\text{diag}(C)$ is a vector formed from the diagonal elements of C; the constraint $\text{diag}(C) = 0$ ensures that a data point does not use itself in its own representation. Although many such matrices C may exist, the ones satisfying $C_{ij} = 0$ if data points x_i and x_j are from different subspaces are of particular interest since they are good candidates for building the affinity matrix. To find such a matrix C, existing methods typically choose a norm $\|\cdot\|$ and solve the regularized optimization problem

$$\min_C \|C\| \ \text{ subject to(s.t.) } \ X = XC \ \text{ and } \ \text{diag}(C) = 0. \tag{1}$$

Once C is obtained, the affinity matrix is usually defined as $|C| + |C^T|$ and spectral clustering is performed. Clearly, the choice of the norm is important to the performance of the method because it affects the type of matrix C that is computed, which subsequently affects the definition of the affinity matrix used during the spectral clustering step. Sparse subspace clustering methods use the ℓ_1-norm to promote sparsity in the coefficient matrix C [13–15,37,38,52], whereas low-rank subspace clustering (LRSC) methods use the nuclear norm to encourage C to be low-rank. Other choices of regularization that have been studied include the ℓ_2-norm [33], TraceLasso norm [32], elastic net regularization [50], and a mixture of ℓ_1-norm and nuclear norm regularization [47]. In this paper, we focus on the class of LRSC methods.

Several LRSC models have been studied. The work in [29,31] presents a convex formulation of LRSC for noiseless data and data with outliers. An alternative framework that builds upon nonconvex formulations of LRSC is presented in [43] for data contaminated by dense noise and sparse gross errors. A latent low-rank representation method is proposed in [30,35,36] for joint subspace clustering and feature extraction. As an alternative low-rank model, [28] proposed the fix-rank representation method. To deal with nonlinear manifolds, a graph Laplacian regularized LRSC is presented in [49]. Finally, a more general discussion on LRSC approaches that includes the relationship between several LRSC methods is presented in [53].

1.2 The alternating direction method of multipliers and proximal gradient method

The alternating direction method of multipliers (ADMM) has found many applications that include robust principle component analysis, consensus optimization, matrix completion, power state estimation, and statistical estimation. The framework was first introduced in the 1970s by Glowinski and Marroco [19] and Gabay and Mercier [18]. The theory of ADMM was further developed by Fortin and Glowinski [16], and its relationship with the Douglas–Rachford splitting procedure for monotone operators [27] was established by Gabay in [17]. Although there is a plethora of papers on ADMM, here we simply highlight the tutorial style papers by Eckstein and Yao [12] and Boyd et al. [5] for the interested reader.

Most analyses for ADMM are in the convex setting, although researchers have recently established convergence results for the nonconvex case [9,22,46,48] under

certain assumptions. The work [22] is particularly relevant, which proves convergence to a first-order solution under assumptions that include the Lipschitz continuity of the gradient of a function appearing in the objective function. Establishing that this assumption holds for the objective function arising in certain LRSC formulations is a key contribution of our manuscript.

Proximal gradient calculations play an important role in the formulation of optimization algorithms. The basic proximal gradient method can be applied to problems whose objective function is the sum of two functions, one that is differentiable and the other that has a closed form solution for an associated proximal mapping. These methods and their variants have been very popular, with notable examples being those designed for sparse [4] and low rank [39] optimization. Although originally designed for convex problems, proximal gradient methods can also be used to solve nonconvex instances [2,6]. The convergence results—similar to ADMM—require a certain Lipschitz continuity assumption on the gradient of one of the problem functions.

1.3 Notation

The matrix $X \in \mathbb{R}^{D\times N}$ is reserved for the data matrix, where D is the dimension of each data point and N is the number of data points; we define $R := \min\{D, N\}$. Given a vector x and positive integer i we let $[x]_i$ denote the i-th component of x. Similarly, for a matrix S and pair of positive integers (i, j) we let $[S]_{ij}$ denote the (i, j)-th entry of S. For real-valued matrices X and Y that are of the same dimension, we let $\langle X, Y\rangle_F := \text{trace}(X^T Y) = \sum_{i,j}[X]_{ij}[Y]_{ij}$ denote the Frobenius inner-product and $\|X\|_F^2 = \langle X, X\rangle_F = \sum_{i,j}[X]_{ij}^2$ the squared Frobenius norm.

1.4 Review of LRSC and contributions of this paper

The main contributions of this paper involve guarantees of converge for the optimization problem used in the LRSC method introduced in [43]. (Henceforth, the acronym LRSC is used to refer to this particular method.) For LRSC, the authors compute the matrix C from an optimization problem different from (1) that aims to model noise that is often present in data. Specifically, the optimization problem used by LRSC to define the affinity matrix C is given by

$$\min_{A,C,E} \|C\|_* + \tfrac{\tau}{2}\|A - AC\|_F^2 + \gamma\|E\|_1 \quad \text{s.t.} \quad X = A + E \ \text{ and } \ C = C^T \tag{2}$$

for chosen parameters $\tau \in (0, \infty)$ and $\gamma \in (0, \infty]$, where $X \in \mathbb{R}^{D\times N}$ is the given data matrix, $E \in \mathbb{R}^{D\times N}$ represents sparse corruptions, and $A \in \mathbb{R}^{D\times N}$ represents the sum of the clean data matrix with dense noise. Since A is assumed to contain dense noise an explicit constraint of the form $A = AC$ that encodes the self-expressiveness property (see the paragraph before (1)) would not be compatible. For this reason, a soft constraint on $A = AC$ is used via a penalty term in (2).

Using the result [43, Theorem 1], we know that for fixed A, the solution to

$$\min_{C} \ \|C\|_* + \tfrac{\tau}{2}\|A - AC\|_F^2 \ \text{ s.t. } \ C = C^T \tag{3}$$

has the closed-form minimizer $C = V\mathcal{P}_\tau(\Sigma)V^T$, where $U\Sigma V^T = A$ is the reduced singular value decomposition (SVD) of A with $U \in \mathbb{R}^{D\times R}$, $V \in \mathbb{R}^{N\times R}$, and $\Sigma \in \mathbb{R}^{R\times R}$, and $\mathcal{P}_\tau$ is the nonlinear thresholding operator defined componentwise as

$$[\mathcal{P}_\tau(\Sigma)]_{ij} := \begin{cases} 1 - \frac{1}{\tau[\Sigma]_{ij}^2} & \text{if } [\Sigma]_{ij} \in \mathcal{I}_1 := (1/\sqrt{\tau}, +\infty), \\ 0 & \text{if } [\Sigma]_{ij} \in \mathcal{I}_2 := [0, 1/\sqrt{\tau}], \end{cases} \tag{4}$$

for all $\{i, j\} \subset \{1, \ldots, R\}$. (We note that we choose to use a reduced SVD of size $R = \min\{D, N\}$ instead of a compact SVD of size $\operatorname{rank}(A)$ to make the analysis performed in Sect. 2 less cumbersome.) Using this result, it is also shown [43, Theorem 1] that the optimal value of the objective in (3) is

$$\Phi_\tau(A) := \sum_{i=1}^{R} \phi_\tau([\Sigma]_{ii}) \geq 0, \tag{5}$$

where $[\Sigma]_{ii}$ is the i-th singular value of A,

$$\phi_\tau(\sigma) := \left(1 - \tfrac{1}{2\tau}\sigma^{-2}\right)\mathbb{1}_{\mathcal{I}_1}(\sigma) + \tfrac{\tau}{2}\sigma^2\mathbb{1}_{\mathcal{I}_2}(\sigma), \tag{6}$$

and the indicator function is defined for an arbitrary set $\mathcal{I}$ as

$$\mathbb{1}_{\mathcal{I}}(\alpha) := \begin{cases} 1 & \text{if } \alpha \in \mathcal{I}, \\ 0 & \text{if } \alpha \notin \mathcal{I}. \end{cases}$$

Therefore, solving problem (2) is equivalent to first solving the optimization problem

$$\min_{A,E} \ \Phi_\tau(A) + \gamma\|E\|_1 \ \text{ s.t. } \ X = A + E \tag{7}$$

for (A_*, E_*) and then setting $C_* = V_*\mathcal{P}_\tau(\Sigma_*)V_*^T$, where $U_*\Sigma_*V_*^T = A_*$ is the reduced SVD of A_*. Solving the nonconvex problem (7) is not straightforward because of the definition of the spectral function Φ_τ. In Sect. 2 we establish critical properties of Φ_τ that will be used to analyze optimization algorithms for solving (7).

The contributions made by this paper include the following:

– We prove that the spectral function Φ_τ in (7), which is used in the definition of the objective function from [43], is differentiable and that its derivative is Lipschitz continuous. Moreover, we are able to give the explicit value for the Lipschitz constant by using several elements of spectral function theory.

- After establishing the smoothness properties of the spectral function, we use them to establish convergence for an ADMM algorithm for solving the optimization problem introduced in [43] for LRSC; this answers an open theoretical question about the convergence of ADMM for this low-rank subspace clustering formulation. Specifically, we prove that the iterate sequence has at least one limit point, and that every limit point satisfies the first-order optimality conditions. This does mean, unfortunately, that limit points could be maximizers or saddle points since the objective function is nonconvex. We also use the derived Lipschitz constant for $\nabla \Phi_\tau$ to give a computable threshold value that the penalty parameter must satisfy in order for the analysis to hold.
- As an alternative to the ADMM, we present and analyze a proximal gradient method for solving the same optimization problem. Even though the optimization problem is nonconvex, we are able to use its structure and our derived explicit value for the Lipschitz constant of $\nabla \Phi_\tau$ to prove that the iterates are uniformly bounded and that all limit points satisfy first-order optimality conditions.
- Numerical experiments illustrate the performance of our approach on real data from face and digit clustering and motion segmentation.

We emphasize that although (2) and (7) have the same minimizers, we analyze the latter because we do not know how to prove convergence of ADMM when applied to (2). (In general, it has been shown [10] that ADMM is not guaranteed to convergence in the 3-block setting, with (2) being such an example.) Thus, we have exchanged problem (2) whose objective function is defined from three simple functions, for formulation (7) whose objective function is defined by one simple function and one more complicated function Φ_τ because we are able to prove a first-order convergence result. As an illustrative example, properties that we establish for Φ_τ are used in Lemma 5 to prove that the objective function in a key subproblem of ADMM is strongly convex for an appropriate and computable choice of the penalty parameter.

1.5 Paper outline

In Sect. 2 we establish critical properties of the spectral function appearing in (7). We use these properties of the spectral function to establish a convergence result for an ADMM algorithm and a proximal gradient method in Sect. 3. We give numerical experiments on real data (motion segmentation, face clustering, and handwritten digits clustering) in Sect. 4. Conclusions are presented in Sect. 5.

2 Properties of the spectral function Φ_τ

The most efficient algorithms for solving problem (7) require Φ_τ to be differentiable. For this reason, our first result proves that Φ_τ is differentiable as well as offers an explicit form of the derivative, which is needed to implement an optimization algorithm.

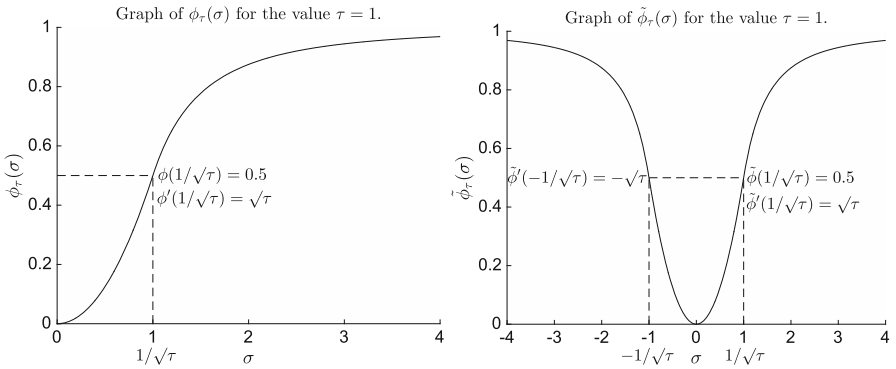


Fig. 2 Graphs of the function ϕ_τ from (5) and the function $\tilde{\phi}_\tau$ used in the proof of Lemma 1 for $\tau = 1$. Other positive values of τ produce functions of a similar shape

Lemma 1 *The spectral function $\Phi_\tau : \mathbb{R}^{D\times N} \to \mathbb{R}$ defined in (5) is continuously differentiable with gradient given by*

$$\nabla\Phi_\tau(A) := U\, diag[\phi'_\tau([\Sigma]_{11}), \ldots, \phi'_\tau([\Sigma]_{RR})]V^T, \tag{8}$$

where $R := \min\{D, N\}$ and $U \in \mathbb{R}^{D\times R}$, $V \in \mathbb{R}^{N\times R}$, and $\Sigma \in \mathbb{R}^{R\times R}$ are the factors of the reduced SVD of A satisfying $U\Sigma V^T = A$.

Proof For any $A \in \mathbb{R}^{D\times N}$ let $R = \min\{D, N\}$ and U, V, and Σ be the factors of a reduced SVD of A as given in the statement of this lemma. According to the definition of Φ_τ in (5), $\Phi_\tau(A)$ can be regarded as a function of the singular values of A given by $\{[\Sigma]_{ii}\}_{i=1}^R$, i.e., $\Phi_\tau(A) = f(\sigma(A))$, where $\sigma(A) \in \mathbb{R}^R$ denotes the vector of singular values of the matrix A and $f : \mathbb{R}^R \to \mathbb{R}$ is

$$f(\sigma) := \sum_{i=1}^{R} \tilde{\phi}_\tau([\sigma]_i), \quad \text{where } \tilde{\phi}_\tau([\sigma]_i) := \begin{cases} \phi_\tau([\sigma]_i) & \text{if } [\sigma]_i \geq 0, \\ \phi_\tau(-[\sigma]_i) & \text{if } [\sigma]_i < 0. \end{cases}$$

See Fig. 2 for an illustration of ϕ_τ and $\tilde{\phi}_\tau$, and note that our introduction of $\tilde{\phi}_\tau$ allows for f to be defined over all of $\mathbb{R}^R$. It is not difficult to show using the definition of ϕ_τ (also see Fig. 2) that all first-order partial derivatives of f exist and are continuous on $\mathbb{R}^R$. It follows that f is continuously differentiable on $\mathbb{R}^R$ and therefore by [26, Corollary 7.4] we also know that Φ_τ is differentiable. The gradient of Φ_τ can be computed by applying [26, Theorem 7.1], which gives the result

$$\nabla\Phi_\tau(A) := U\, \mathrm{diag}[\phi'_\tau([\Sigma]_{11}), \ldots, \phi'_\tau([\Sigma]_{RR})]V^T, \tag{9}$$

which is the desired result (8). This completes the proof. □

Our next aim is to prove that $\nabla\Phi_\tau$ is Lipschitz continuous whenever ϕ' is Lipschitz continuous, and that in this case their Lipschitz constants are the same. Our proof requires two preliminary results and a definition.

Lemma 2 ([1, Proposition 4.1]) *Let $f : \mathbb{R}_+ \to \mathbb{R}$ be any Lipschitz continuous function satisfying $f(0) = 0$, $L(f)$ denote the smallest Lipschitz constant of f, i.e.,*

$$L(f) := \sup_{x,y\in\mathbb{R}_+} \frac{|f(x) - f(y)|}{|x - y|},$$

and define

$$L_{\mathbb{T}}(f) := \sup_{x,y\in\mathbb{R}_+,c\in\mathbb{T}} \frac{|f(x) - cf(y)|}{|x - cy|},$$

where $\mathbb{T} := \{z \in \mathbb{C} \mid |z| = 1\}$. Then, it follows that $L_{\mathbb{T}}(f) \equiv L(f)$.

The main result of this section uses a result related to complex doubly substochastic matrices. The definition of this subset of matrices is given next.

Definition 1 *(Complex doubly substochastic)* A complex square matrix is called complex doubly substochastic if and only if the ℓ_1-norm of each row and column is less than or equal to 1.

With this definition, we may now state the next known result.

Lemma 3 ([1, Lemma 3.1]) *Define the set of permutation functions of size R by*

$$\Pi := \{\pi \mid \pi : \{1, \ldots, R\} \to \{1, \ldots, R\} \text{ is a bijection}\}$$

and the set of vectors in $\mathbb{C}^R$ with unimodular entries as

$$\mathcal{U} := \{u \in \mathbb{C}^R \mid |[u]_i| = 1 \text{ for all } 1 \le i \le R\}.$$

It holds that a matrix $A \in \mathbb{C}^{R\times R}$ is complex doubly substochastic if and only if

$$A \in conv(\{M_{\pi,u} \mid \pi \in \Pi \text{ and } u \in \mathcal{U}\})$$

where $M_{\pi,u} \in \mathbb{R}^{R\times R}$ is defined as

$$[M_{\pi,u}]_{i,j} := \begin{cases} [u]_i & \text{if } j = \pi(i), \\ 0 & \text{otherwise,} \end{cases}$$

for all $\{i, j\} \subset \{1, \ldots, R\}$ and $conv(\mathcal{S})$ denotes the convex hull of the set $\mathcal{S}$.

We can now establish a sufficient condition for the Lipschitz continuity of $\nabla\Phi_\tau$. We remark that the proof is based on the logic used to establish [1, Theorem 1.1].

Lemma 4 *Let $\nabla\Phi_\tau(A)$ be defined as in Lemma 1. If ϕ'_τ is Lipschitz continuous, then $\nabla\Phi_\tau$ is Lipschitz continuous and they have the same Lipschitz constants.*

Proof Let $A \in \mathbb{R}^{D\times N}$ and $B \in \mathbb{R}^{D\times N}$ be arbitrary matrices and let us write their reduced SVD factorizations as

$$U_A \Sigma_A V_A^T = A \text{ and } U_B \Sigma_B V_B^T = B, \tag{10}$$

where U_A and U_B are in $\mathbb{R}^{D\times R}$, Σ_A and Σ_B are in $\mathbb{R}^{R\times R}$, and V_A and V_B are in $\mathbb{R}^{N\times R}$. It follows that the quantity $\|A-B\|_F^2$ can be written as

$$\begin{aligned}
\|A-B\|_F^2 &= \|A\|_F^2 + \|B\|_F^2 - 2\langle A, B\rangle_F \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\left\langle U_A\Sigma_A V_A^T, U_B\Sigma_B V_B^T\right\rangle_F \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\left\langle U_B^T U_A\Sigma_A, \Sigma_B V_B^T V_A\right\rangle_F \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\sum_{i,j}\left[(\Sigma_B V_B^T V_A)\odot(U_B^T U_A\Sigma_A)\right]_{ij},
\end{aligned} \tag{11}$$

where $\odot$ denotes the Hadamard product. It can be shown that $(V_B^T V_A)\odot(U_B^T U_A)$ is complex doubly substochastic by first showing that the two-norm of each row and column of $V_B^T V_A$ and $U_B^T U_A$ is less than or equal to 1 (this latter property can be shown using the orthogonality of the columns of the matrices). Therefore, Lemma 3 implies that there exists a positive integer m, scalars $\{c_1,\ldots,c_m\}\subset[0,1]$ satisfying $\sum_{\ell=1}^m c_\ell = 1$, permutation functions $\{\pi_\ell\}_{\ell=1}^m$ of length R, and a set of vectors $\{u_\ell\}_{\ell=1}^m \subset \mathbb{C}^R$ such that $|[u_\ell]_p| = 1$ for all $1\le\ell\le m$ and $1\le p\le R$, satisfying

$$(V_B^T V_A)\odot(U_B^T U_A) = \sum_{\ell=1}^m c_\ell M_{\pi_\ell,u_\ell}. \tag{12}$$

Combining (11) with $(\Sigma_B V_B^T V_A)\odot(U_B^T U_A\Sigma_A) \equiv \Sigma_B(V_B^T V_A \odot U_B^T U_A)\Sigma_A$, (12), the fact that $\sum_{i,j}[\Sigma_B(\sum_{\ell=1}^m c_\ell M_{\pi_\ell,u_\ell})\Sigma_A]_{ij}$ is in $\mathbb{R}$ in light of (12), $\sum_{\ell=1}^m c_\ell = 1$, the definition of M_{π_ℓ,u_ℓ}, and $|[u_\ell]_p| = 1$ shows that

$$\begin{aligned}
\|A-B\|_F^2 &= \|A\|_F^2 + \|B\|_F^2 - 2\sum_{i,j}\left[\Sigma_B\left(\sum_{\ell=1}^m c_\ell M_{\pi_\ell,u_\ell}\right)\Sigma_A\right]_{ij} \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\mathrm{Re}\left(\sum_{i,j}\left[\Sigma_B\left(\sum_{\ell=1}^m c_\ell M_{\pi_\ell,u_\ell}\right)\Sigma_A\right]_{ij}\right) \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\mathrm{Re}\left(\sum_{\ell=1}^m c_\ell \sum_{i,j}\left[\Sigma_B M_{\pi_\ell,u_\ell}\Sigma_A\right]_{ij}\right) \\
&= \sum_{\ell=1}^m c_\ell\left[\sum_{p=1}^R [\Sigma_A]_{pp}^2 + \sum_{p=1}^R [\Sigma_B]_{pp}^2 - 2\mathrm{Re}\left(\sum_{p=1}^R [\Sigma_B]_{pp}[u_\ell]_p[\Sigma_A]_{[\pi_\ell]_p[\pi_\ell]_p}\right)\right] \\
&= \sum_{\ell=1}^m c_\ell \sum_{p=1}^R \left|[\Sigma_B]_{pp} - [u_\ell]_p[\Sigma_A]_{[\pi_\ell]_p[\pi_\ell]_p}\right|^2,
\end{aligned} \tag{13}$$

where $Re(c)$ denotes the real part of a complex number c.

Next, we recall from (8) that $\nabla\Phi_\tau(A)$ and $\nabla\Phi_\tau(B)$ have the same form as (10) except for the singular values are different. Since m, $\{c_\ell\}_{\ell=1}^m$, $\{u_\ell\}_{\ell=1}^m$, and $\{\pi_\ell\}_{\ell=1}^m$ do not depend on Σ_A and Σ_B, using the same argument as above for $\nabla\Phi_\tau(A)$ and $\nabla\Phi_\tau(B)$ in place of A and B and using (8) shows that

$$\|\nabla\Phi_\tau(A) - \nabla\Phi_\tau(B)\|_F^2 = \sum_{\ell=1}^{m} c_\ell \sum_{p=1}^{R} \left|\phi_\tau'([\Sigma_B]_{pp}) - [u_\ell]_p \phi_\tau'([\Sigma_A]_{[\pi_\ell]_p,[\pi_\ell]_p})\right|^2.$$

Combining this equality with the assumption that ϕ_τ' is Lipschitz continuous, the definition of $L_{\mathbb{T}}(\phi_\tau')$ in Lemma 2, and (13) yields

$$\begin{aligned}\|\nabla\Phi_\tau(A) - \nabla\Phi_\tau(B)\|_F^2 &\le \left(L_{\mathbb{T}}(\phi_\tau')\right)^2 \sum_{\ell=1}^{m} c_\ell \sum_{p=1}^{R} \left|[\Sigma_B]_{pp} - [u_\ell]_p [\Sigma_A]_{[\pi_\ell]_p[\pi_\ell]_p}\right|^2 \\ &= \left(L_{\mathbb{T}}(\phi_\tau')\right)^2 \|A - B\|_F^2 = \left(L(\phi_\tau')\right)^2 \|A - B\|_F^2,\end{aligned}$$

which is the desired conclusion. This completes the proof. □

We have arrived to the main result of this section.

Theorem 1 *The function $\nabla\Phi_\tau$ is Lipschitz continuous with constant $L(\nabla\Phi_\tau) := 3\tau$.*

Proof It is straightforward to show that $\phi_\tau'(\sigma) = \frac{1}{\tau}\sigma^{-3}\mathbb{1}_{\mathcal{I}_1}(\sigma) + \tau\sigma\mathbb{1}_{\mathcal{I}_2}(\sigma)$ is Lipschitz continuous with constant 3τ. Thus, Lemma 4 gives the desired result. □

Now that we know how to compute $\nabla\Phi_\tau(A)$ (see Lemma 1) and that $\nabla\Phi_\tau$ is Lipschitz continuous with constant $L(\nabla\Phi_\tau) = 3\tau$, we proceed to discuss algorithms that use these properties to solve our key optimization problem (7).

3 Algorithms for solving the optimization problem (7)

We now consider two algorithms for solving (7). Specifically, in Sect. 3.1 we describe an ADMM method and in Sect. 3.2 we describe a proximal gradient method.

3.1 An ADMM algorithm

An ADMM algorithm for solving (7) is given as Algorithm 1. The method is based on the augmented Lagrangian (AL) function for problem (7), which is defined as

$$\mathcal{L}(A, E, Y) := \Phi_\tau(A) + \gamma\|E\|_1 + \langle Y, X - A - E\rangle_F + \tfrac{\rho}{2}\|X - A - E\|_F^2 \qquad (14)$$

with $\rho \in (0, \infty)$ the penalty parameter and $Y \in \mathbb{R}^{D\times N}$ the matrix of dual variables. The basic idea behind ADMM is to perform one pass of alternating minimization over

E and A (for fixed Y_k), followed by a first-order multiplier update to Y_k. These updates should be performed in sequence during the kth iteration and take the form

$$\begin{aligned} E_{k+1} &= \operatorname{argmin}_E \ \gamma\|E\|_1 + \langle Y_k, X - A_k - E\rangle_F + \tfrac{\rho}{2}\|X - A_k - E\|_F^2 \\ &= \operatorname{argmin}_E \ \frac{\gamma}{\rho}\|E\|_1 + \tfrac{1}{2}\left\|\tfrac{1}{\rho}Y_k + X - A_k - E\right\|_F^2 \\ &= \operatorname{prox}_{\frac{\gamma}{\rho}\|\cdot\|_1}\left(\tfrac{1}{\rho}Y_k + X - A_k\right) \qquad (15) \\ A_{k+1} &= \operatorname{argmin}_A \ \Phi_\tau(A) + \langle Y_k, X - A - E_{k+1}\rangle_F + \tfrac{\rho}{2}\|X - A - E_{k+1}\|_F^2 \\ &= \operatorname{argmin}_A \ \tfrac{1}{\rho}\Phi_\tau(A) + \tfrac{1}{2}\left\|\tfrac{Y_k}{\rho} + X - A - E_{k+1}\right\|_F^2 \\ &= \operatorname{prox}_{\frac{1}{\rho}\Phi_\tau}\left(\tfrac{1}{\rho}Y_k + X - E_{k+1}\right) \qquad (16) \\ Y_{k+1} &= Y_k + \rho(X - A_{k+1} - E_{k+1}), \qquad (17) \end{aligned}$$

where $\operatorname{prox}_f(Z) := \operatorname{argmin}_B f(B) + \frac{1}{2}\|B - Z\|_F^2$ denotes the proximal operator of f.

For any $\epsilon > 0$, the proximal operator of $\epsilon\|\cdot\|_1$, namely $\operatorname{prox}_{\epsilon\|\cdot\|_1}(Z)$, is given by the shrinkage thresholding operator $\mathcal{S}_\epsilon(Z)$, which is defined as

$$[\mathcal{S}_\epsilon(Z)]_{ij} := \operatorname{sgn}([Z]_{ij}) \cdot \max(|[Z]_{ij}| - \epsilon, 0). \qquad (18)$$

Therefore, the matrix E_{k+1} that solves (15) and that is computed in Step 3 of Algorithm 1 corresponds to $\epsilon = \gamma/\rho$ and may be calculated as

$$E_{k+1} = S_{\frac{\gamma}{\rho}}\left(\tfrac{1}{\rho}Y_k + X - A_k\right). \qquad (19)$$

For any $\epsilon > 0$, the computation of the proximal operator of $\epsilon\Phi_\tau$ is more involved. In fact, it is shown in [43, Theorem 3] that one can exploit the fact that Φ_τ is a spectral function to reduce the computation of $\operatorname{prox}_{\epsilon\Phi_\tau}$ to the computation of $\operatorname{prox}_{\epsilon\phi_\tau}$, where ϕ_τ is defined in (6). Specifically, if $Z = U\Sigma V^T$ is the SVD of Z, then $\operatorname{prox}_{\epsilon\Phi_\tau}(Z) = U\Lambda V^T$, where each diagonal entry of Λ is obtained from the corresponding diagonal entry of Σ as $[\Lambda]_{ii} = \operatorname{prox}_{\epsilon\phi_\tau}([\Sigma]_{ii})$. Now, the quantity $\operatorname{prox}_{\epsilon\phi_\tau}(\sigma)$ for any $\sigma \geq 0$ is given by the argument that solves the optimization problem

$$\min_\lambda \ \phi(\lambda; \sigma) \qquad (20)$$

where

$$\phi(\lambda;\sigma) := \tfrac{1}{2}(\sigma - \lambda)^2 + \epsilon\phi_\tau(\lambda) = \tfrac{1}{2}(\sigma - \lambda)^2 + \epsilon\begin{cases} 1 - \frac{1}{2\tau}\lambda^{-2} & \text{if } \lambda > 1/\sqrt{\tau} \\ \frac{\tau}{2}\lambda^2 & \text{if } \lambda \leq 1/\sqrt{\tau}. \end{cases} \qquad (21)$$

The first-order optimality condition for the problem in (20) is given by the system

$$\sigma = \psi(\lambda), \quad \text{with } \psi(\lambda) := \begin{cases} \lambda + \frac{\epsilon}{\tau}\lambda^{-3} & \text{if } \lambda > 1/\sqrt{\tau}, \\ \lambda + \epsilon\tau\lambda & \text{if } \lambda \leq 1/\sqrt{\tau}. \end{cases} \qquad (22)$$

Since the system in (22) is equivalent to a system of polynomial equations in λ, to compute $\text{prox}_{\epsilon\phi_\tau}(\sigma)$, we need to find a root λ of this system that minimizes the function $\phi(\cdot\,;\sigma)$ in (20). A method for computing such a root is described in the proof of [43, Theorem 4] and formalized here as Algorithm 2. Specifically, Step 3 of Algorithm 2 computes $\lambda = \text{prox}_{\epsilon\phi_\tau}(\sigma)$ for $\sigma = [\Sigma]_{ii}$, $\lambda = [\Lambda]_{ii}$ and $\epsilon = 1/\rho$. The choice $\epsilon = 1/\rho$ corresponds to the computation of the matrix A_{k+1} that solves (16) and that is computed in Step 4 of Algorithm 1 as

$$A_{k+1} = U\Lambda V^T, \quad \frac{Y_k}{\rho} + X - E_{k+1} = U\Sigma V^T, \quad \text{and} \quad [\Lambda]_{ii} = \text{prox}_{\epsilon\phi_\tau}([\Sigma]_{ii}). \tag{23}$$

Algorithm 1 ADMM for solving (7).

1: Choose values $A_0 \in \mathbb{R}^{D\times N}$, $Y_0 \in \mathbb{R}^{D\times N}$, and $\rho > 6\tau$ (see the proof of Lemma 8).
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute E_{k+1} from (15) for given (A_k, Y_k), i.e., set E_{k+1} according to (19).
4: Compute A_{k+1} from (16) for given (E_{k+1}, Y_k), i.e., set A_{k+1} according to Algorithm 2.
5: Compute Y_{k+1} from (17) for given (E_{k+1}, A_{k+1}).
6: **end for**

Algorithm 2 Algorithm for computing A_{k+1} that solves (16).

1: Compute a reduced SVD of the matrix $X - E_{k+1} + \frac{1}{\rho}Y_k$ such that $U\Sigma V^T = X - E_{k+1} + \frac{1}{\rho}Y_k$.
2: Set $\Lambda \in \mathbb{R}^{R\times R}$ to be the zero matrix.
3: **for** $i = 1, \ldots, R$ **do**
4: **if** $3\tau \le \rho$ **then**
5: **if** $[\Sigma]_{ii} \le (\rho+\tau)/(\rho\sqrt{\tau})$ **then** ▷ see Fig. 3a
6: Set $[\Lambda]_{ii} \leftarrow \left(\frac{\rho}{\rho+\tau}\right)[\Sigma]_{ii}$.
7: **else**
8: Set $[\Lambda]_{ii}$ as the unique real number satisfying $[\Lambda]_{ii} > 1/\sqrt{\tau}$ and $[\Sigma]_{ii} = \psi([\Lambda]_{ii})$.
9: **end if**
10: **else**
11: **if** $[\Sigma]_{ii} < \frac{4}{3}\sqrt[4]{\frac{3}{\rho\tau}}$ **then** ▷ see Fig. 3b
12: Set $[\Lambda]_{ii} \leftarrow \left(\frac{\rho}{\rho+\tau}\right)[\Sigma]_{ii}$.
13: **else if** $[\Sigma]_{ii} > \frac{\rho+\tau}{\rho\sqrt{\tau}}$ **then**
14: Set $[\Lambda]_{ii}$ as the unique real number satisfying $[\Lambda]_{ii} > 1/\sqrt{\tau}$ and $[\Sigma]_{ii} = \psi([\Lambda]_{ii})$.
15: **else**
16: Set $[\Lambda^{(1)}]_{ii} \leftarrow \left(\frac{\rho}{\rho+\tau}\right)[\Sigma]_{ii}$. ▷ note that $[\Lambda^{(1)}]_{ii} \in (0, 1/\sqrt{\tau}]$
17: Set $[\Lambda^{(2)}]_{ii}$ as the larger of the two $\Lambda \in \mathbb{R}$ satisfying $\Lambda > 1/\sqrt{\tau}$ and $[\Sigma]_{ii} = \psi(\Lambda)$.
18: **if** $\phi([\Lambda^{(1)}]_{ii}; [\Sigma]_{ii}) \le \phi([\Lambda^{(2)}]_{ii}; [\Sigma]_{ii})$ **then**
19: Set $[\Lambda]_{ii} \leftarrow [\Lambda^{(1)}]_{ii}$.
20: **else**
21: Set $[\Lambda]_{ii} \leftarrow [\Lambda^{(2)}]_{ii}$.
22: **end if**
23: **end if**
24: **end if**
25: **end for**
26: Set $A_{k+1} \leftarrow U\Lambda V^T$.

A convergence guarantee for Algorithm 1 is not immediate because problem (7) is nonconvex. Our goal for the remainder of this section is to use the properties we

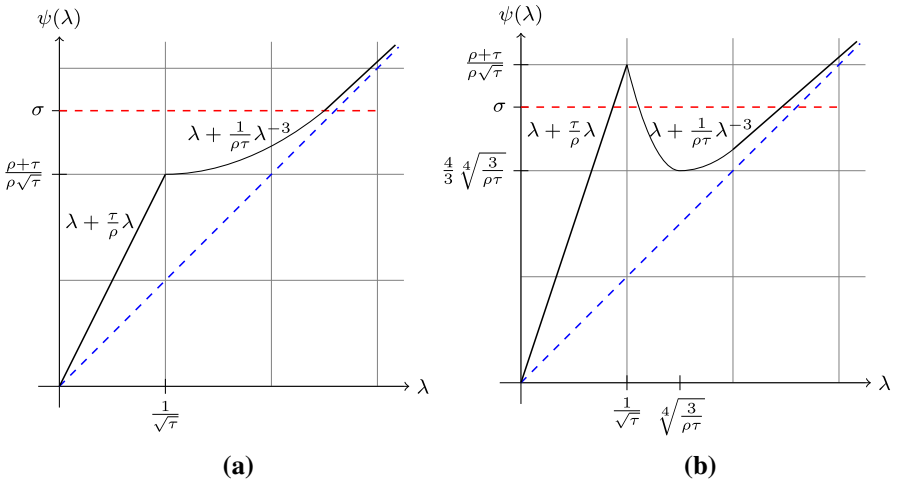


Fig. 3 Plots of $\psi(\lambda)$ with $\epsilon = 1/\rho$ defined in (22) for different values of τ and ρ. (**a**) Plot of $\psi(\lambda)$ when $3\tau \leq \rho$, (**b**) plot of $\psi(\lambda)$ when $3\tau > \rho$

established in Sect. 2 for Φ_τ to show that the assumptions in [22, Section 4] hold, which will then allow us to deduce that Algorithm 1 has a convergence guarantee.

We first show that the minimization problem in (16) is strongly convex if $\rho > 3\tau$.

Lemma 5 *For all $k \geq 0$, the function $\mathcal{L}(A, E_{k+1}, Y_k)$ being minimized in (16) is strongly convex with strong convexity constant $\rho - 3\tau$.*

Proof Showing that the function $\mathcal{L}(A, E_{k+1}, Y_k)$ being minimized in (16) is strongly convex is equivalent to showing that $\Phi_\tau(A) + \frac{\rho}{2}\|A\|_F^2$ is strongly convex. From the Lipschitz continuity of $\nabla\Phi_\tau$ (see Theorem 1) and [11, Theorem 3.1.4] it follows that for any $\{A_1, A_2\} \subset \mathbb{R}^{D\times N}$ the following inequality holds:

$$|\Phi_\tau(A_1) - \Phi_\tau(A_2) - \langle\nabla\Phi_\tau(A_2), A_1 - A_2\rangle_F| \leq \frac{L(\nabla\Phi_\tau)}{2}\|A_1 - A_2\|_F^2. \quad (24)$$

We can use the definition of the absolute value and rearrange terms in (24) to obtain

$$\Phi_\tau(A_1) \geq \Phi_\tau(A_2) + \langle\nabla\Phi_\tau(A_2), A_1 - A_2\rangle_F - \frac{L(\nabla\Phi_\tau)}{2}\|A_1 - A_2\|_F^2, \quad (25)$$

and then add $\frac{\rho}{2}\|A_1\|_F^2$ to both sides to obtain

$$\begin{aligned}
&\Phi_\tau(A_1) + \tfrac{\rho}{2}\|A_1\|_F^2 \\
&\geq \Phi_\tau(A_2) + \langle\nabla\Phi_\tau(A_2), A_1 - A_2\rangle_F - \tfrac{L(\nabla\Phi_\tau)}{2}\|A_1 - A_2\|_F^2 + \tfrac{\rho}{2}\|A_1\|_F^2 \\
&= \Phi_\tau(A_2) + \tfrac{\rho}{2}\|A_2\|_F^2 + \langle\nabla\Phi_\tau(A_2) + \rho A_2, A_1 - A_2\rangle_F \\
&\quad - \tfrac{\rho}{2}\|A_2\|_F^2 - \langle\rho A_2, A_1 - A_2\rangle_F - \tfrac{L(\nabla\Phi_\tau)}{2}\|A_1 - A_2\|_F^2 + \tfrac{\rho}{2}\|A_1\|_F^2 \\
&= \Phi_\tau(A_2) + \tfrac{\rho}{2}\|A_2\|_F^2 + \langle\nabla\Phi_\tau(A_2) + \rho A_2, A_1 - A_2\rangle_F + \tfrac{\rho - L(\nabla\Phi_\tau)}{2}\|A_1 - A_2\|_F^2 \\
&= \Phi_\tau(A_2) + \tfrac{\rho}{2}\|A_2\|_F^2 + \langle\nabla\Phi_\tau(A_2) + \rho A_2, A_1 - A_2\rangle_F + \tfrac{\rho - 3\tau}{2}\|A_1 - A_2\|_F^2,
\end{aligned}$$

where we used Theorem 1 in the last equality. The previous inequality gives the desired result once we note that $\rho > 6\tau > 3\tau$ by the choice of ρ in Algorithm 1. □

We next establish a critical bound on the distance between consecutive Lagrange multiplier estimates produced by the multiplier update (17).

Lemma 6 *For all $k \geq 1$, the Lagrange multiplier estimates satisfy*

$$\|Y_{k+1} - Y_k\|_F \leq 3\tau \|A_{k+1} - A_k\|_F.$$

Proof Lemma 5 established that the objective function in problem (16) is strongly convex so that the matrix A_{k+1} exists and is unique. Moreover, from the optimality conditions for problem (16) we know that A_{k+1} must satisfy

$$\nabla \Phi_\tau(A_{k+1}) - Y_k - \rho(X - A_{k+1} - E_{k+1}) = 0.$$

Combining this with (17) yields

$$\nabla \Phi_\tau(A_{k+1}) = Y_{k+1} \tag{26}$$

for all $k \geq 0$, which may then be combined with Theorem 1 to deduce that

$$\|Y_{k+1} - Y_k\|_F = \|\nabla \Phi_\tau(A_{k+1}) - \nabla \Phi_\tau(A_k)\|_F \leq 3\tau \|A_{k+1} - A_k\|_F \tag{27}$$

for all $k \geq 1$, which completes the proof. □

We now show that the augmented Lagrangian is bounded below over the sequence of iterates computed by Algorithm 1.

Lemma 7 *For all $k \geq 1$, it holds that*

$$\mathcal{L}(A_k, E_k, Y_k) \geq \gamma \|E_k\|_1 + \Phi_\tau(X - E_k) + \tfrac{\rho - 3\tau}{2} \|X - E_k - A_k\|_F^2 \geq 0.$$

Proof It follows from the definition of $\mathcal{L}$, (26), (24) with $A_1 = X - E_k$ and $A_2 = A_k$, Theorem 1, and (5) that the following holds for all $k \geq 1$:

$$\begin{aligned}
&\mathcal{L}(A_k, E_k, Y_k) \\
&= \gamma \|E_k\|_1 + \Phi_\tau(A_k) + \langle Y_k, X - E_k - A_k \rangle + \tfrac{\rho}{2} \|X - E_k - A_k\|_F^2 \\
&= \gamma \|E_k\|_1 + \Phi_\tau(A_k) + \langle \nabla \Phi_\tau(A_k), X - E_k - A_k \rangle + \tfrac{\rho}{2} \|X - E_k - A_k\|_F^2 \\
&\geq \gamma \|E_k\|_1 + \Phi_\tau(X - E_k) + \tfrac{\rho - L(\nabla \phi_\tau)}{2} \|X - E_k - A_k\|_F^2 \\
&= \gamma \|E_k\|_1 + \Phi_\tau(X - E_k) + \tfrac{\rho - 3\tau}{2} \|X - E_k - A_k\|_F^2 \geq 0,
\end{aligned}$$

where the final claim of non-negativity follows from $\rho > 6\tau > 3\tau$ by the choice of ρ in Algorithm 1. This is the desired result and therefore completes the proof. □

Our next lemma summarizes the critical properties of the iterates computed by the ADMM algorithm that are needed to prove convergence.

Lemma 8 *The sequence* $\{(A_k, E_k, Y_k)\}_{k\geq 0}$ *computed by the ADMM method stated as Algorithm 1 satisfies the following properties:*

D1 *The iterates* A_{k+1} *and* E_{k+1} *exist and are unique. Also, the sequence of augmented Lagrangian values* $\{\mathcal{L}(A_k, E_k, Y_k)\}_{k\geq 0}$ *is uniformly bounded below.*

D2 *For the constant* $c := (3\tau)^2$ *it holds that*

$$\|Y_{k+1} - Y_k\|_F^2 \leq c\big(\|A_{k+1} - A_k\|_F^2 + \|E_{k+1} - E_k\|_F^2\big).$$

D3 *The feasible set for problem (7) is nonempty,* Φ_τ *is smooth, and* $\gamma\|E\|_1$ *is convex.*

D4 *Problem (15) is strongly convex with parameter* $c_E := \rho$ *and problem (16) is strongly convex with parameter* $c_A := \rho - 3\tau > 0$. *Also, these constants satisfy*

$$\rho c_E \geq 2c \quad \text{and} \quad \rho c_A \geq 2c$$

where c *is the constant defined in D2.*

Proof To prove D2, we observe from Lemma 6 that, for all $k \geq 1$, we have

$$\|Y_{k+1} - Y_k\|_F^2 \leq (3\tau)^2\|A_{k+1} - A_k\|_F^2 \leq c\big(\|A_{k+1} - A_k\|_F^2 + \|E_{k+1} - E_k\|_F^2\big),$$

where we have used $c = (3\tau)^2$ as defined in the statement of D2.

Next, we prove D4. First, note that problem (15) is clearly strongly convex with parameter $c_E := \rho$ and that problem (16) is strongly convex because of Lemma 5 with parameter $c_A := \rho - 3\tau > 0$. We now use $\rho > 6\tau$ (see Algorithm 1) to derive

$$\begin{aligned}\rho c_E &= \rho^2 > 36\tau^2 > 18\tau^2 = 2(3\tau)^2 = 2c \quad \text{and}\\ \rho c_A &= \rho(\rho - 3\tau) > (6\tau)(6\tau - 3\tau) > 2(3\tau)^2 = 2c,\end{aligned}$$

which completes the proof of D4.

To prove D1, we first observe that the iterates A_{k+1} and E_{k+1} exist and are unique because of the strong convexity noted in the previous paragraph. Combining this with the fact that $\{\mathcal{L}(A_k, E_k, Y_k)\}_{k\geq 0}$ was shown to be bounded uniformly from below in Lemma 7 establishes that D1 holds.

Finally, D3 holds because the feasible set for problem (7) is clearly nonempty, Φ_τ is smooth (see Lemma 1), and $\gamma\|E\|_1$ is a convex function because $\gamma > 0$. □

We now state our convergence result for the ADMM method.

Theorem 2 *The sequence* $\{(A_k, E_k, Y_k)\}_{k\geq 0}$ *computed by the ADMM method stated as Algorithm 1 has at least one limit point, and every one of those limit points satisfies the first-order optimality conditions for problem (7).*

Proof Since D1–D4 in Lemma 8 are precisely conditions D1–D4 from [22, Section 4], we can apply their theory to conclude that every limit point of $\{A_k, E_k, Y_k\}_{k\geq 0}$ satisfies the first-order optimality conditions for problem (7). The fact that at least one limit point exists follows from the fact that the sequence $\{(A_k, E_k, Y_k)\}_{k\geq 0}$ is bounded as shown in [23, Lemma 6.5.6]. This completes the proof. □

3.2 A proximal gradient algorithm

We now describe a proximal gradient based method for solving (7). To this end, we use the constraint in (7) to substitute $E = X - A$ into the objective function of (7), which transforms problem (7) into the unconstrained problem

$$\min_{A} \ \Phi_\tau(A) + \gamma \|A - X\|_1. \tag{28}$$

The proximal gradient algorithm for problem (28) is given by Algorithm 3. Note that $\alpha \in (0, 2/(3\tau)) \equiv (0, 2/L(\nabla\Phi_\tau))$ is a step size parameter that must be chosen and that the minimizer needed in Step 3 has the closed-form solution

$$A_{k+1} = \mathcal{S}_{\alpha\gamma}\left(A_k - \alpha\nabla\Phi_\tau(A_k) - X\right) + X,$$

where $\mathcal{S}_{\alpha\gamma}$ is the shrinkage thresholding operator defined in (18).

Algorithm 3 Proximal gradient algorithm for solving (28).

1: Choose an initial iterate $A_0 \in \mathbb{R}^{D\times N}$ and a step size parameter $\alpha \in (0, 2/(3\tau))$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Compute $A_{k+1} = \operatorname{argmin}_A \ \frac{1}{2\alpha}\|A - (A_k - \alpha\nabla\Phi_\tau(A_k))\|_F^2 + \gamma\|A - X\|_1$.
4: **end for**

Most of the next result is standard for proximal gradient methods. One enhancement, namely that the iterates $\{A_k\}$ are guaranteed to be bounded, is achieved by using the fact that Φ_τ is bounded below by zero as noted in (5).

Lemma 9 *The sequence $\{A_k\}_{k\geq 0}$ from Algorithm 3 is bounded and it has a limit point that satisfies the first-order optimality conditions for problem* (28)*.*

Proof It is known from the theory for proximal gradient methods that if the stepsize parameter is chosen to satisfy $\alpha \in (0, 2/L(\nabla\Phi_\tau))$ (see Algorithm 3 and recall that $L(\nabla\Phi_\tau) = 3\tau$) then the objective function

$$f(A) := \Phi_\tau(A) + \gamma\|A - X\|_1$$

is monotonically decreasing. Thus, it follows that

$$f(A_k) \leq f(A_0) \ \text{ for all } k \geq 0. \tag{29}$$

Now, for a proof by contradiction, assume there exists a subsequence $\mathcal{K}$ of the iterations such that $\{\|A_k\|_F\}_{k\in\mathcal{K}} \to \infty$. Combining this with the observation that $\Phi_\tau(A) \geq 0$ for all A (see (5)), we can conclude that

$$\begin{aligned}\liminf_{k\in\mathcal{K}} f(A_k) &= \liminf_{k\in\mathcal{K}} \left[\Phi_\tau(A_k) + \gamma\|A_k - X\|_1\right] \\ &\geq \liminf_{k\in\mathcal{K}} \Phi_\tau(A_k) + \liminf_{k\in\mathcal{K}} \gamma\|A_k - X\|_1 \\ &\geq \liminf_{k\in\mathcal{K}} \gamma\|A_k - X\|_1 = \infty,\end{aligned}$$

which contradicts (29). Thus, the sequence $\{A_k\}_{k\geq 0}$ is bounded, which means that there exists at least one limit point. The fact that every limit point satisfies the first-order optimality conditions for problem (28) follows from the fact that Φ_τ is a continuously differentiable nonconvex function with Lipschitz continuous gradient (see Lemma 1 and Lemma 4) and [2,6]. This completes the proof. □

4 Numerical results

In this section, we evaluate the performance of the LRSC method. We refer to the instance of the LRSC method that uses Algorithm 1 as the subproblem solver as LRSC-ADMM and the one based on using Algorithm 3 as the subproblem solver as LRSC-PROX. To evaluate these approaches, we compare their performance to the state-of-the-art low rank subspace clustering methods LRR [31] and REDU-EXPR [53]. The LRR method computes the affinity matrix C as a solution to the optimization problem

$$\min_{C,E} \|C\|_* + \gamma\|E\|_{2,1} \;\text{ s.t. }\; X = XC + E, \tag{30}$$

for some chosen parameter $\gamma \in (0, \infty)$, where $\|E\|_{2,1}$ is the sum of the ℓ_2 norms of the columns of E. The REDU-EXPR first applies robust PCA [8] to the data X by computing a denoised data L by solving the optimization problem

$$\min_{L,E} \|L\|_* + \gamma\|E\|_1 \;\text{ s.t. }\; X = L + E, \tag{31}$$

and then computes the affinity matrix C from L by setting it to $C = V_1 V_1^T$, where $U_1\Sigma_1 V_1^T = L$ is the reduced SVD of L.

For all methods tested, once a coefficient matrix, say C^*, has been found, we use $|C^*| + |C^*|^T$ as the symmetric affinity matrix. This affinity matrix is used in the spectral clustering phase as mentioned in the first paragraph of Sect. 1.1. Since spectral clustering involves solving a nonconvex optimization problem associated with k-means, we perform 20 clustering trials with random initializations and select the one with the lowest k-means objective function value. Once this clustering has been obtained, we compute the clustering accuracy as

$$\text{clustering accuracy} = \frac{\#\text{ of correctly classified data points}}{\text{total }\#\text{ of data points}}.$$

All experiments are run on a single core of an Intel Xeon E5-2680v3 (Haswell) processor with 128 GB DDR4 RAM and all algorithms are written in MATLAB.

An implementation of our solvers must include termination conditions and choices for algorithm parameters. Since LRSC-ADMM and LRSC-PROX are designed to solve the optimization problem (7), our termination conditions should reflect this fact. In particular, we terminate when the triple (A_k, E_k, Y_k) approximately satisfies the first-order necessary optimality conditions for problem (7), namely, when it satisfies

$$\frac{\|X - A_k - E_k\|_1}{DN} \leq 10^{-9}, \tag{32a}$$

$$\frac{\|\nabla \Phi_\tau(A_k) - Y_k\|_1}{DN} \leq 10^{-9}, \quad \text{and} \tag{32b}$$

$$\inf_{G \in \partial\gamma\|E_k\|_1} \frac{\|Y_k - G\|_1}{DN} \leq 10^{-9}, \tag{32c}$$

or 500 iterations is reached. Any triple (A_k, E_k, Y_k) satisfying (32) with 10^{-9} replaced by 0 is called a Karush–Kuhn Tucker (KKT) point for problem (7). The iterate triple (A_k, E_k, Y_k) is automatically obtained for LRSC-ADMM directly in Algorithm 1. On the other hand, for LRSC-PROX only the iterate A_k is obtained in Algorithm 3. However, motivated by the optimality conditions in (32a) and (32b), for LRSC-PROX we set (for each A_k) $E_k = X - A_k$ and $Y_k = \nabla \Phi_\tau(A_k)$ during each iteration. LRSC-ADMM is initialized with $(A_0, Y_0) = (X, \nabla \Phi_\tau(X))$ and the parameter value $\rho = 7\tau$ was used. LRSC-PROX was initialized with $A_0 = X$ and the parameter value $\alpha = 1/(6\tau)$ was used. In both cases, the parameter value was chosen based on obtaining the best optimization performance.

4.1 Results for motion segmentation

Motion segmentation refers to the problem of decomposing a video that contains multiple rigid-moving objects into multiple regions that correspond to the different motions. This problem is often solved by first extracting feature points (see Fig. 1) that are tracked throughout F frames of the video. Then, the set of coordinates corresponding to each feature point are vectorized to form a data point of dimension $2F$ obtained by concatenating the (x, y) coordinates for each frame. Finally, the collection of data points are clustered so that ideally each cluster will correspond to a single moving object. This is achieved by exploiting the fact that the set of trajectories associated with a single rigid moving object lie approximately in an affine subspace of dimension at most 3. Therefore, the trajectories of several rigid moving objects lie approximately in a union of affine subspaces. Since LRSC is designed to cluster linear subspaces, we append the constant 0.1 to each trajectory vector and work with the $2F+1$ dimensional data vectors.

Table 1 Clustering accuracy (as a percentage), computational time (in seconds), and number of iterations averaged over all videos from the Hopkins155 database

# of motions	LRR	REDU-EXPR	LRSC-ADMM	LRSC-PROX
Accuracy				
2	95.26	93.24	96.82	96.82
3	90.78	89.90	93.02	93.02
Time				
2	7.48	2.49	0.27	0.11
3	15.18	3.11	0.47	0.16
# of iterations				
2	327.5	500.0	21.9	17.7
3	333.1	500.0	26.0	23.6

We use the Hopkins155 [40] database, which contains 155 videos (120 videos with 2 motions and 35 videos with 3 motions). For each video, trajectories are extracted automatically with a tracking algorithm implemented in OpenCV [34] and the ground truth segmentation is obtained by manually labeling the first frame and eliminating wrong trajectories, as described in [40]. Table 1 reports the average clustering accuracy ("accuracy"), average computational time ("time"), and average number of iterations required by the solver ("# of iterations")—here, the average is over all of the videos for a given number of motions. Parameter tuning for all algorithms gave the following: (i) for LRR we use $\gamma = 100$ and 150 for 2 and 3 motions, respectively; (ii) for REDU-EXPR we use $\gamma = 0.05$ and 0.06 for 2 and 3 motions, respectively; and (iii) for LRSC-ADMM and LRSC-PROX we use $(\gamma, \tau) = (5, 250)$ for 2 motions and $(\gamma, \tau) = (5, 350)$ for 3 motions. From Table 1 we can observe that LRSC-ADMM and LRSC-PROX achieve the best clustering accuracy, while at the same time requiring the smallest computational time. Although not presented here, we also verified that the final objective function values for LRSC-ADMM and LRSC-PROX differed by a relative value of at most 10^{-6}. Finally, both LRSC-ADMM and LRSC-PROX require relatively few iterations to satisfy the termination conditions in (32).

4.2 Results on face clustering

Face clustering is the problem of clustering a set of face images of multiple individuals according to the identity of each individual. For a Lambertian object, the set of all images taken from the same viewpoint and with the same expression but under different lighting conditions lies approximately in a low dimensional subspace [3]. Moreover, due to cast shadows and specularities, a few pixels of the face image can have large errors. Therefore, the face clustering problem can be treated as a subspace clustering problem with data corrupted by small noise and sparse gross errors.

We use the Extended Yale B database [25], which includes 64 frontal face images of 38 individuals acquired under 64 different lighting conditions. To reduce the computational cost, we downsample the original images to 48 by 42 pixels and treat each

Table 2 Clustering accuracy (as a percentage), average total computational time (in seconds), and number of iterations averaged over 20 trials on the face clustering data

# of subjects	LRR	REDU-EXPR	LRSC-ADMM	LRSC-PROX
Accuracy				
2	97.40	89.91	98.15	98.15
10	67.35	72.44	65.33	65.33
38	72.44	71.36	70.52	70.52
Time				
2	4.5	14.5	1.0	0.2
10	106.0	243.9	82.0	32.6
38	3867.0	4676.4	2900.3	1787.8
# of iterations				
2	239.3	500	1.0	1.0
10	243.7	500	73.6	35.7
38	239.0	500	212.0	136.0

2016-dimensional vectorized image as a data point. Since LRSC is designed to cluster linear subspaces, we follow the setup in [43] and append the constant 0.03 to each vectorized image; this gives a 2017-dimensional feature vector for each image. We apply the four methods to cluster $n \in \{2, 10, 38\}$ subjects and record in Table 2 the average clustering accuracy ("accuracy"), average computational time ("time"), and average number of iterations required by the solver ("# of iterations") over 20 trials—each trial consists of choosing n subjects randomly from among the 38 subjects. The problem parameters that we use for the different methods are tuned based on the clustering performance and are given by the following: (i) for LRR we set $\gamma = 0.005$, 0.003, and 0.0075 to correspond to $n = 2$, 10, and 38, respectively; (ii) for REDU-EXPR we set $\gamma = 0.011$, 0.01, and 0.01 to correspond to $n = 2$, 10, and 38, respectively; and (iii) for both LRSC-ADMM and LRSC-PROX we set $(\gamma, \tau) = (0.05, 0.03)$ for $n = 2$, $(\gamma, \tau) = (0.02, 0.05)$ for $n = 10$, and $(\gamma, \tau) = (0.0075, 0.065)$ for $n = 38$.

Table 2 shows that both LRR, LRSC-ADMM, and LRSC-PROX achieve the best clustering accuracies for the 2 subject setting. For the 10 subject case, LRR, LRSC-ADMM, and LRSC-PROX again achieve similar clustering accuracies, but they are surpassed by REDU-EXPR by approximately 6%. Finally, for the 38 subject case the 4 methods achieve clustering accuracies within 2% of each other with LRR being the best at 72.44%. In terms of computational time, LRSC-ADMM and LRSC-PROX again are the most efficient by a significant margin in all cases. As was true in the previous section, both LRSC-ADMM and LRSC-PROX require relatively few iterations to satisfy the termination conditions in (32). Although not presented here, we also verified that the final objective function values for LRSC-ADMM and LRSC-PROX differed by a relative value of at most 10^{-6}. We note that LRSC-ADMM and LRSC-PROX only require 1 iteration for the 2 subject case because the initialization $A_0 = X$, $E_0 = 0$, and $Y_0 = \nabla \Phi_\tau(X)$ is approximately optimal.

Table 3 Clustering accuracy (as a percentage), computational time (in seconds), and number of iterations averaged over 20 trials on the handwritten digits data

# of images	LRR	REDU-EXPR	LRSC-ADMM	LRSC-PROX
Accuracy				
500	78.98	77.93	80.44	80.48
2000	80.14	79.99	80.08	80.08
5000	79.68	80.84	81.77	81.77
Time				
500	44.0	54.4	8.9	4.2
2000	1473.0	184.5	2.1	3.3
5000	24,633.5	518.4	10.3	18.3
# of iterations				
500	274.0	500.0	49.6	11.8
2000	186.7	500.0	1.0	1.0
5000	184.9	500.0	1.0	1.0

4.3 Results on clustering images of handwritten digits

The MNIST database [24] contains 70,000 images of handwritten digits 0–9. We test the performance of the LRSC methods on clustering a set of images drawn from MNIST into 10 groups corresponding to the 10 digits. Following [51], we represent each image by a feature vector of dimension 3,472 computed from the scattering transform [7]. Such feature vectors are translation invariant and deformation stable.

In each experiment, we randomly choose $N/10$ images for each digit 0–9, where $N \in \{500, 2000, 5000\}$, and then apply LRSC to cluster the resulting N images. To reduce the computational cost, the feature vectors for all images in each experiment are projected to dimension 500 using PCA before performing LRSC. The average clustering accuracy ("accuracy"), average computational time ("time"), and average number of iterations required by the solver ("# of iterations") for $N \in \{500, 2000, 5000\}$ are reported in Table 3—the average is over 20 random choices of the $N/10$ images. Tuning the parameters for each method resulted in the following: (i) for LRR we use $\gamma = 0.006$; (ii) for REDU-EXPR we use $\gamma = 0.001$; and (iii) for both LRSC-ADMM and LRSC-PROX we use $(\gamma, \tau) = (0.02, 0.01)$. From Table 3 we may observe that all four methods achieve similar clustering accuracies, and consistent with the previous sections LRSC-ADMM and LRSC-PROX require the smallest computational time by a wide margin. Although not presented, we verified that the final objective values for LRSC-ADMM and LRSC-PROX differed by a relative value of at most 10^{-6}. Finally, consistent with the previous sections, we can observe that relatively few iterations are required by LRSC-ADMM and LRSC-PROX to satisfy the termination conditions in (32). In particular, we comment that LRSC-ADMM and LRSC-PROX only need 1 iteration for the 2000 and 5000 images cases for the same reason as described in the last paragraph of Sect. 4.2.

5 Conclusions

We established convergence of an ADMM algorithm for solving the optimization problem introduced in [43] for LRSC, thus answering an open theoretical question for this LRSC formulation. As an alternative to the ADMM, we presented and analyzed a proximal gradient method for solving the same optimization problem. Our numerical experiments illustrated that our method on real data from face and digit clustering and motion segmentation is comparable (often better) than state-of-the-art LRSC methods in terms of clustering accuracy. (An exception is face clustering in the 10 subject case since our methods obtained 7% less accuracy.) In terms of efficiency, our ADMM and proximal gradient method both consistently required the smallest computational time by a wide margin, with neither always being superior to the other.

Acknowledgements The authors thank the financial support of NSF Grants 1447822, 1618637, and 1704458.

References

1. Andersson, F., Carlsson, M., Perfekt, K.M.: Operator-Lipschitz estimates for the singular value functional calculus. In: Proceedings of the American Mathematical Society (2015)
2. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss–Seidel methods. Math. Program. **137**(1–2), 91–129 (2013)
3. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Anal. Mac. Intel. **25**(2), 218–233 (2003)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
5. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends®. Mach. Learn. **3**(1), 1–122 (2011)
6. Bredies, K., Lorenz, D.A., Reiterer, S.: Minimization of non-smooth, non-convex functionals by iterative thresholding. J. Optim. Theory Appl. **165**(1), 78–112 (2015)
7. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
8. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 11 (2011)
9. Chartrand, R., Wohlberg, B.: A nonconvex ADMM algorithm for group sparsity with sparse groups. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6009–6013. IEEE (2013)
10. Chen, C., He, B., Ye, Y., Yuan, X.: The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. Math. Program. **155**(1–2), 57–79 (2016)
11. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2000)
12. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. Technical Report Center for Operations Research, Rutgers University, 640 Bartholomew Road, Piscataway, New Jersey (2015)
13. Elhamifar, E., Vidal, R.: Clustering disjoint subspaces via sparse representation. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1926–1929. IEEE (2010)
14. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2797. IEEE (2009)
15. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2765–2781 (2013)

16. Fortin, M., Glowinksi, R.: On decomposition-coordination methods using an augmented Lagrangian. In: Fortin, M., Glowinksi, R. (eds.) Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems. Chapter 3, pp. 97–144. Elsevier, Amsterdam (1983)
17. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinksi, R. (eds.) Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems, pp. 299–331. Elsevier, Amsterdam (1983). Chapter 9
18. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Comput. Math. Appl. **2**(1), 17–40 (1976)
19. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique **9**(R2), 41–76 (1975). http://eudml.org/doc/193269
20. Ho, J., Yang, M.H., Lim, J., Lee, K.C., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 11–18. IEEE (2003)
21. Hong, W., Wright, J., Huang, K., Ma, Y.: Multiscale hybrid linear models for lossy image representation. IEEE Trans. Image Process. **15**(12), 3655–3671 (2006)
22. Hong, M., Luo, Z.Q., Razaviyayn, M.: Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. SIAM J. Optim. **26**(1), 337–364 (2016)
23. Jiang, H.: Augmented Lagrangian based algorithms for nonconvex optimization with applications in subspace clustering. Ph.D. thesis, Johns Hopkins University (2016). https://jscholarship.library.jhu.edu/handle/1774.2/838
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
25. Lee, K.C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
26. Lewis, A.S., Sendov, H.S.: Nonsmooth analysis of singular values. part I: Theory. Set-Valued Anal. **13**(3), 213–241 (2005)
27. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**(6), 964–979 (1979). https://doi.org/10.1137/0716071
28. Liu, R., Lin, Z., De la Torre, F., Su, Z.: Fixed-rank representation for unsupervised visual learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 598–605. IEEE (2012)
29. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning, pp. 663–670 (2010)
30. Liu, G., Yan, S.: Latent low-rank representation for subspace segmentation and feature extraction. In: 2011 IEEE International Conference on Computer Vision, pp. 1615–1622. IEEE (2011)
31. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 171–184 (2013)
32. Lu, C., Lin, Z., Yan, S.: Correlation adaptive subspace segmentation by trace lasso. In: IEEE International Conference on Computer Vision, pp. 1345–1352 (2013)
33. Lu, C.Y., Min, H., Zhao, Z.Q., Zhu, L., Huang, D.S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: European Conference on Computer Vision, pp. 347–360 (2012)
34. Open source computer vision library (OpenCV). http://sourceforge.net/projects/opencvlibrary
35. Patel, V.M., Nguyen, H.V., Vidal, R.: Latent space sparse subspace clustering. In: International Conference on Computer Vision (2013)
36. Patel, V.M., Nguyen, H., Vidal, R.: Latent space sparse and low-rank subspace clustering. IEEE J. Sel. Top. Signal Process. **9**(4), 691–701 (2015)
37. Soltanolkotabi, M., Candes, E.J.: A geometric analysis of subspace clustering with outliers. The Annals of Statistics, pp. 2195–2238 (2012)
38. Soltanolkotabi, M., Elhamifar, E., Candes, E.: Robust subspace clustering. Ann. Stat. **42**(2), 669–699 (2014)
39. Toh, K.C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac. J. Optim. **6**(615–640), 15 (2010)
40. Tron, R., Vidal, R.: A benchmark for the comparsion of 3-D motion segmentation algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
41. Vidal, R., Ma, Y., Sastry, S.: Generalized Principal Component Analysis. Springer Interdisciplinary Applied Mathematics (2016)

42. Vidal, R.: Subspace clustering. IEEE Signal Process. Mag. **28**(2), 52–68 (2010)
43. Vidal, R., Favaro, P.: Low rank subspace clustering (LRSC). Pattern Recognit. Lett. **43**, 47–61 (2014)
44. Vidal, R., Tron, R., Hartley, R.: Multiframe motion segmentation with missing data using PowerFactorization and GPCA. Int. J. Comput. Vis. **79**(1), 85–105 (2008)
45. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
46. Wang, F., Cao, W., Xu, Z.: Convergence of multi-block Bregman ADMM for nonconvex composite problems (2015). arXiv preprint arXiv:1505.03063
47. Wang, Y.X., Xu, H., Leng, C.: Provable subspace clustering: When LRR meets SSC. In: Neural Information Processing Systems (2013)
48. Wang, Y., Yin, W., Zeng, J.: Global convergence of ADMM in nonconvex nonsmooth optimization (2015). arXiv preprint arXiv:1511.06324
49. Yin, M., Gao, J., Lin, Z.: Laplacian regularized low-rank representation and its applications. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 504–517 (2016)
50. You, C., Li, C.G., Robinson, D., Vidal, R.: Oracle based active set algorithm for scalable elastic net subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3928–3937 (2016)
51. You, C., Robinson, D., Vidal, R.: Scalable sparse subspace clustering by orthogonal matching pursuit. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3918–3927 (2016)
52. You, C., Vidal, R.: Geometric conditions for subspace-sparse recovery. In: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp. 1585–1593 (2015)
53. Zhang, H., Lin, Z., Zhang, C., Gao, J.: Relations among some low-rank subspace recovery models. Neural Comput. **27**(9), 1915–1950 (2015)