**ℓ₀-Motivated Low-Rank Sparse Subspace Clustering**

Maria Brbić and Ivica Kopriva, Senior Member, IEEE

**Abstract**—In many applications, high-dimensional data points can be well represented by low-dimensional subspaces. To identify the subspaces, it is important to capture a global and local structure of the data which is achieved by imposing low-rank and sparseness constraints on the data representation matrix. In low-rank sparse subspace clustering (LRRSSC), nuclear and ℓ₁-norms are used to measure rank and sparsity. However, the use of nuclear and ℓ₁-norms leads to an over penalized problem and only approximates the original problem. In this paper, we propose two ℓ₀ quasi-norm-based regularizations. First, this paper presents regularization based on multivariate generalization of minimax-concave penalty (GMC-LRRSSC), which contains the global minimizers of a ℓ₀ quasi-norm regularized objective. Afterward, we introduce the Schatten-0 (S₀) and ℓ₀-regularized objective and approximate the proximal map of the joint solution using a proximal average method (S₀/ℓ₀-LRRSSC). The resulting nonconvex optimization problems are solved using an alternating direction method of multipliers with established convergence conditions of both algorithms. Results obtained on synthetic and four real-world datasets show the effectiveness of GMC-LRRSSC and S₀/ℓ₀-LRRSSC when compared to state-of-the-art methods.

**Index Terms**—Alternating direction method of multipliers (ADMMs), generalization of the minimax-concave (GMC) penalty, ℓ₀ regularization, low-rank, sparsity, subspace clustering.

**I. INTRODUCTION**

HIGH-DIMENSIONAL data analysis is a widespread problem in many applications of machine learning, computer vision, and bioinformatics [1]–[6]. However, in many real-world datasets, the intrinsic dimension of high-dimensional data is much smaller than the dimension of the ambient space and data can be well represented as lying close to a union of low-dimensional subspaces. The problem of segmenting data according to the low-dimensional subspaces they are drawn from is known as subspace clustering [7]. Thanks to their capability to handle arbitrarily shaped clusters and their well-defined mathematical principles, spectral-based methods [8], [9] are widely used approaches to subspace clustering. These methods solve the subspace clustering problem by relying on the spectral graph theory and cluster eigenvectors of the graph Laplacian matrix corresponding to the smallest eigenvalues [10].

One of the main challenges in subspace clustering is the construction of the affinity matrix that captures well (dis)similarities between data points. Among various approaches proposed in the literature, methods based on sparse representation [11], [12] and low-rank representation (LLR) [13]–[15] have been among the most successful in many applications [16]. These methods exploit the self-expressiveness property of the data and represent each data point as a linear combination of other data points in the dataset. LRR [13], [14], [17] captures the global structure of the data by imposing a low-rank constraint on the data representation matrix. Low rank implies that the representation matrix is described by a weighted sum of a small number of outer products of left and right singular vectors. In order to ensure convexity of the related optimization problem, the rank minimization is relaxed as the nuclear or Schatten-1 norm minimization problem [18]–[20]. Different from LRR, sparse subspace clustering (SSC) [11], [21] captures local linear relationships by constraining the representation matrix to be sparse. Using the tightest convex relaxation of the ℓ₀ quasi-norm, the SSC model solves the sparsity maximization problem as the ℓ₁-norm minimization problem [22], [23]. Both LRR and SSC guarantee exact clustering when subspaces are independent, but the independence assumption is overly restrictive for many real-world datasets [24], [25]. Under appropriate conditions [26], SSC also succeeds for disjoint subspaces. However, when the number of dimensions is higher than three, SSC can face connectivity problems, resulting in a disconnected graph within a subspace [27]. A natural way to construct an adaptive model that is able to capture the global and the local structure of the data is to constrain the representation matrix to be low rank and sparse. In [16] and [28]–[30], that is done by combining nuclear and ℓ₁-norms as the measures of rank and sparsity, respectively. The motivation lies in the fact that minimization of these norms results in a convex optimization problem.

Although convex, nuclear and ℓ₁-norms are not exact measures of rank and sparsity. Therefore, the optimal solution of the nuclear and ℓ₁-norms regularized objective is only an approximate solution of the original problem [31]. A proximity operator associated with the nuclear norm penalizes large singular values, leading to biased results in low-rank-constrained optimization problems [32], [33]. Similarly, in sparsity regularized problems, the ℓ₁-norm solution systematically underestimates high amplitude components of sparse vectors [34]. Nonconvex regularizations based on ℓ_p quasi-norms (0 ≤ p < 1) or their approximations have been...
proposed for various low-rank [32], [33], [35]–[39] and sparsity regularized problems [34], [40]–[47]. Recently, nonconvex approximations of rank and sparsity have also been introduced in the subspace clustering problem [48]–[52]. Specifically, \( \ell_0 \)-induced SSC is introduced in [48]. The corresponding optimization problem is solved using the proximal gradient descent which under assumptions on the sparse eigenvalues converges to a critical point. Jiang et al. [49] replaced the nuclear norm regularizer with the nonconvex Ky Fan \( p-k \) norm [53] and proposed the proximal iteratively reweighted optimization algorithm to solve the problem. In [50] and [54], the rank is approximated by using Schatten-\( q \) quasi-norm regularization (\( 0 < q < 1 \)). The optimization problem in [50] is solved using the generalized matrix soft thresholding algorithm [55]. Schatten-\( q \) quasi-norm minimization with tractable \( q = 2/3 \) and \( q = 1/2 \) is proposed in [51]. The Schatten-\( q \) (\( S_q \)) quasi-norm for \( 0 < q < 1 \) is equivalent to the \( \ell_q \) quasi-norm on a vector of singular values. Compared to the nuclear norm, it makes a closer approximation of the rank function. In this regard, the \( S_q \) quasi-norm can be seen as an equivalent to the \( \ell_0 \) quasi-norm and stands for the definition of the rank function. Furthermore, [52] combined the \( S_q \) regularizer (\( 0 < q < 1 \)) for low-rank and the \( \ell_p \) quasi-norm regularizer (\( 0 < p < 1 \)) for sparsity constraints. However, recent results in [56] show that in \( \ell_p \)-regularized least squares (\( 0 \leq p < 1 \)), smaller values of \( p \) lead to more accurate solutions. If the \( \ell_1 \)-norm is also considered, the authors show that for large measurement noises, \( \ell_1 \) outperforms \( \ell_p \), \( p < 1 \). However, for small measurement noises, \( \ell_0 \) quasi-norm regularization outperforms \( \ell_p \), \( 0 < p \leq 1 \).

Motivated by the limitations discussed above, we introduce two \( S_0/\ell_0 \) quasi-norm-based nonconvex regularizations for low-rank SSC (LRSSC). First, we propose regularization based on multivariate generalization of the minimax-concave (GMC) penalty function, introduced in [34] for sparsity-regularized linear least squares. Here, this approach is extended to the rank approximation. The GMC penalty enables us to maintain the convexity of low-rank and sparsity-constrained subproblems, while achieving better approximation of rank and sparsity than nuclear and \( \ell_1 \)-norms. Importantly, this regularization is closely related to the continuous exact \( \ell_0 \) penalty which contains the global minimizers of the \( \ell_0 \) quasi-norm regularized least-squares objective [34], [57]. GMC penalty yields solutions of low-rank and sparsity-constrained subproblems based on firm thresholding of singular values and coefficients of the representation matrix, respectively. The firm thresholding function \( H : \mathbb{R} \to \mathbb{R} \) is defined as [59]

\[
H(x; \lambda) = \begin{cases} 
 x, & \text{if } |x| > \sqrt{2 \lambda} \\
 0, & \text{if } |x| \leq \sqrt{2 \lambda} \\
 0, & \text{if } |x| < \frac{1}{2 \lambda}.
\end{cases}
\]

Simultaneous rank and sparsity regularization are handled by using the proximal average method, introduced in [62] for convex problems and extended recently to nonconvex and non-smooth functions [63], [64]. Proximal average allows us to approximate the proximal map of the joint solution by averaging solutions obtained separately from low-rank and sparsity subproblems, leading to a problem with low computational cost in each iteration. Furthermore, using the proximal average method enables us to establish the global convergence guarantee for \( S_0/\ell_0 \)-regularized LRSSC.

Better approximation of rank and sparsity is a consequence of the properties of firm and hard thresholding operators associated with GMC and \( \ell_0 \) regularizations. As opposed to them, the soft thresholding operator underestimates high amplitude coefficients in the \( \ell_1 \)-norm-based sparsity regularized objective, as well as large singular values in the low-rank approximation problem. As an example, Fig. 1 shows soft, firm, and hard thresholding operators used in LRSSC, GMC-LRSSC, and \( S_0/\ell_0 \)-LRSSC, respectively.

To solve the corresponding optimization problems, we derive algorithms based on the computationally efficient alternating direction method of multipliers (ADMMs) [65]. Although ADMM has been successfully applied to many nonconvex problems [66]–[68], only recent theoretical results establish convergence of ADMM for certain nonconvex functions [69]–[72]. For GMC regularization, we show that the sequence generated by the algorithm is bounded and prove that any limit point of the iteration sequence is a stationary point. For \( S_0/\ell_0 \) regularization with a proximal average approach, based on the property that \( \ell_0 \) and \( S_0 \) quasi-norms belong to a class of semialgebraic functions and satisfy the Kurdyka–Łojasiewicz inequality [73], the global convergence of the algorithm can be guaranteed.

Experimental results on synthetic and four real-world datasets demonstrate that the proposed \( \ell_0 \)-based LRSSC algorithms converge fast and to a point with lower or similar clustering error (CE) than the convex approximations with
nuclear and $\ell_1$-norms. Compared to the state-of-the-art subspace clustering methods, the proposed algorithms perform better on four benchmark datasets.

A. Contributions

The contributions of this paper are summarized as follows.

1) We introduce the nonconvex generalized minimax-concave penalty in the LRSSC problem such that the exact $\ell_0$ penalty [57]. The introduced penalty maintains the convexity of the sparsity and low-rank constrained subproblems. The proximal operator of the related GMC penalty function is the firm thresholding function [34].

2) We introduce $S_0$ and $\ell_0$ pseudo-norm regularizations for LRSSC. Using the proximal average method [62], [63], we average the solutions of proximal maps of low-rank and sparsity subproblems, with the hard thresholding function as a proximity operator of the related penalties [59].

3) We derive ADMM-based optimization algorithms for LRSSC constrained either with a GMC penalty or with $S_0/\ell_0$ quasi-norms. Iterative firm or hard thresholding of singular values and coefficients of the representation matrix is used to obtain the solution of rank and sparsity-constrained subproblems.

4) We prove that the sequence generated by the GMC-regularized LRSSC algorithm is bounded and that any limit point of the iteration sequence is a stationary point that satisfies Karush–Kuhn–Tucker (KKT) conditions.

5) We establish the convergence property of the $S_0/\ell_0$ regularized approach with proximal average and show that the algorithm converges regardless of the initialization. To the best of our knowledge, we are the first to show convergence with $S_0$ and $\ell_0$ penalties in the low-rank and sparsity-constrained optimization problem.

The remainder of this paper is organized as follows. Section II gives a brief overview of the related work. In Sections III and IV, we introduce GMC and $S_0/\ell_0$ regularized LRSSC methods, respectively. We formulate the problem, present optimization algorithms, and analyze the convergence and computational complexity. The experimental results on synthetic and four real-world datasets are presented in Section V. Finally, Section VI concludes this paper.

B. Main Notation

Scalars are denoted by lowercase letters, vectors are denoted by bold lowercase letters, matrices are denoted by bold capital letters, and subspaces are denoted by calligraphic letters. $\| \cdot \|_F$ denotes the Frobenius norm defined as the square root of the sum of the squares of matrix elements. $\| \cdot \|_1$ denotes the $\ell_1$-norm defined as the sum of absolute values of matrix elements. $\| \cdot \|_s$ denotes the nuclear norm defined as the sum of singular values of a matrix. The $\ell_0$ quasi-norm is denoted by $\| \cdot \|_0$ and for matrix $A \in \mathbb{R}^{n \times M}$ defined as

$$\|A\|_0 = \# \{ a_{ij} \neq 0, \; i = 1 \ldots N, \; j = 1 \ldots M \}$$

where $\#$ denotes the cardinality function. The Schatten-0 quasi-norm is denoted by $\| \cdot \|_{S_0}$ and defined as

$$\|A\|_{S_0} = \| \text{diag}(A) \|_0$$

where $A = U \Sigma V^T$ is the singular value decomposition (SVD) of matrix $A$. Since the $\ell_0$ quasi-norm does not satisfy the homogeneous property, it is not a norm, but with a slight abuse of notation, we will refer to it as the $\ell_0$-norm in the rest of this paper. The null vector is denoted by $0$, and $	ext{diag}()$ is the vector of diagonal elements of a matrix. Table I summarizes some notations used in this paper.

| Notation | Definition |
|----------|------------|
| $N$      | Number of data points |
| $n$      | Dimension of data points |
| $L$      | Number of subspaces |
| $X \in \mathbb{R}^{n \times N}$ | Data matrix |
| $C \in \mathbb{R}^{N \times N}$ | Representation matrix |
| $W \in \mathbb{R}^{N \times N}$ | Affinity matrix |
| $X = U \Sigma V^T$ | Singular value decomposition of $X$ |
| $\sigma(X)$ | Vector of singular values of $X$ |

where $\Sigma$ is the diagonal matrix containing the singular values. Let $X_i$ be a submatrix of $X$ of rank $d_i$, $0 < d_i < N$, and $\sum_{i=1}^L N_i = N$. Given data matrix $X$, subspace clustering segments data points according to the low-dimensional subspaces. The first step is the construction of the affinity matrix $W \in \mathbb{R}^{N \times N}$, whose elements represent the similarity between data points. An ideal affinity matrix is block diagonal (up to a permutation); a nonzero distance is assigned to the points in the same subspace and zero distance to the points from different subspaces. The spectral clustering algorithm [8], [9] is then applied to the affinity matrix to obtain memberships of data points to the subspaces.

II. Background

Consider the data matrix $X \in \mathbb{R}^{n \times N}$ the columns of which are data points drawn from a union of $L$ linear subspaces $\bigcup_{i=1}^L S_i$ of unknown dimensions $d_i = \dim(S_i)$ in $\mathbb{R}^n$. Let $X_i \in \mathbb{R}^{n \times N_i}$ be a submatrix of $X$ of rank $d_i$, $0 < d_i < N_i$, and $\sum_{i=1}^L N_i = N$. Given data matrix $X$, subspace clustering segments data points according to the low-dimensional subspaces. The first step is the construction of the affinity matrix $W \in \mathbb{R}^{N \times N}$, whose elements represent the similarity between data points. An ideal affinity matrix is block diagonal (up to a permutation); a nonzero distance is assigned to the points in the same subspace and zero distance to the points from different subspaces. The spectral clustering algorithm [8], [9] is then applied to the affinity matrix to obtain memberships of data points to the subspaces.

A. Related Work

LRR [13], [14] aims to find an LRR matrix $C \in \mathbb{R}^{N \times N}$ for input data matrix $X$ by solving the following convex optimization problem:

$$\min_{C} \|C\|_* \quad \text{s.t.} \quad X = XC$$

where the nuclear norm is used to approximate the rank of $C$. Let $X = U \Sigma V^T$ be the SVD of $X$. The closed-form solution of problem (3) is given by [14]

$$\hat{C} = VV^T.$$  

(4)

When data points are contaminated by additive white Gaussian noise (AWGN), the following minimization problem is solved:

$$\min_{C} \frac{\lambda}{2} \|X - XC\|_F^2 + \|C\|_*$$

(5)
where \( \lambda \) is the rank regularization constant. The optimal solution of problem (5) is given by [17] and [74]

\[
\hat{C} = V_1 \left( I - \frac{1}{\lambda} \Sigma_{-1} \right) V_1^T
\]

(6)

where \( U = [U_1 \ U_2], \Sigma = \text{diag}(\Sigma_1 \ \Sigma_2), \) and \( V = [V_1 \ V_2]. \) Matrices are partitioned according to the sets \( I_1 = \{ i : \sigma_i > (1/\sqrt{\lambda}) \} \) and \( I_2 = \{ i : \sigma_i \leq (1/\sqrt{\lambda}) \}, \) where \( \sigma_i \) denotes the \( i \)th singular value of \( X. \)

SSC [11] represents each data point as a sparse linear combination of other data points and solves the following convex optimization problem:

\[
\min_C \| C \|_1 \quad \text{s.t.} \quad X = XC, \ \text{diag}(C) = 0
\]

(7)

where constraint \( \text{diag}(C) = 0 \) is used to avoid the trivial solution of representing a data point as a linear combination of itself.

For data contaminated by the AWGN, the following minimization problem is solved to approximate sparse representation matrix \( C: \)

\[
\min_C \frac{1}{2} \| X - XC \|_F^2 + \tau \| C \|_1 \quad \text{s.t.} \quad \text{diag}(C) = 0
\]

(8)

where \( \tau \) is the sparsity regularization constant. This problem can be solved efficiently using the ADMM optimization procedure [11], [65].

LRSSC [29] requires that the representation matrix \( C \) be simultaneously low rank and sparse. LRSSC solves the following problem:

\[
\min_C \lambda \| C \|_* + \tau \| C \|_1 \quad \text{s.t.} \quad X = XC, \ \text{diag}(C) = 0
\]

(9)

where \( \lambda \) and \( \tau \) are rank and sparsity regularization constants, respectively. For the AWGN corrupted data, the following problem needs to be solved to approximate \( C: \)

\[
\min_C \frac{1}{2} \| X - XC \|_F^2 + \lambda \| C \|_* + \tau \| C \|_1
\]

\text{s.t.} \ \text{diag}(C) = 0.

(10)

After representation matrix \( C \) is estimated, the affinity matrix \( W \in \mathbb{R}^{N \times N} \) is calculated as follows:

\[
W = |C| + |C|^T.
\]

(11)

In the next two sections, we introduce two nonconvex regularizers for the LRSSC. We formulate the LRSSC problem in the following general form:

\[
\min_C \frac{1}{2} \| X - XC \|_F^2 + g(C) + \tau f(C)
\]

\text{s.t.} \ \text{diag}(C) = 0

(12)

where \( g(C) \) and \( f(C) \) are functions that, respectively, measure the rank and sparsity of the data representation matrix \( C. \) The convex formulation used in (10) implies \( g(C) = \| C \|_* \) and \( f(C) = \| C \|_1. \)

### III. GMC-LRSSC Algorithm

#### A. Problem Formulation

We propose to regularize rank and sparsity using the multivariate GMC penalty function, introduced in [34] for sparse regularized least squares. We start with some definitions and results that will be used throughout this paper.

**Definition 1** [34]: Let \( z \in \mathbb{R}^N \) and \( B \in \mathbb{R}^{M \times N}. \) The GMC penalty function \( \psi_B : \mathbb{R}^N \rightarrow \mathbb{R} \) is defined as

\[
\psi_B(z) = \| z \|_1 - S_B(z)
\]

(13)

where \( S_B : \mathbb{R}^N \rightarrow \mathbb{R} \) is the generalized Huber function defined as

\[
S_B(z) = \inf_{v \in \mathbb{R}^N} \left\{ \| v \|_1 + \frac{1}{2} \| B(z - v) \|_2^2 \right\}.
\]

(14)

**Lemma 1** [34]: Let \( z \in \mathbb{R}^N, y \in \mathbb{R}^M, A \in \mathbb{R}^{M \times N}, \) and \( \lambda > 0. \) Define \( F : \mathbb{R}^N \rightarrow \mathbb{R} \) as

\[
F(z) = \frac{1}{2} \| y - Az \|_2^2 + \lambda \psi_B(z)
\]

(15)

where \( \psi_B : \mathbb{R}^N \rightarrow \mathbb{R} \) is the GMC penalty. If \( A^T A - \lambda B^T B \) is a positive semidefinite matrix, \( F \) is a convex function. The convexity condition is satisfied by setting

\[
B = \sqrt{\gamma/\lambda} A, \quad 0 \leq \gamma \leq 1.
\]

(16)

The parameter \( \gamma \) controls the nonconvexity of the penalty \( \psi_B. \) Larger values of \( \gamma \) increase the nonconvexity of the penalty. The \( \ell_1 \)-norm can be seen as a special case of this penalty by setting \( \gamma = 0. \)

**Lemma 2** [34]: Let \( z \in \mathbb{R}^N, y \in \mathbb{R}^M, A \in \mathbb{R}^{M \times N}, \) and \( \lambda > 0. \) If \( A^T A \) is diagonal with positive entries and \( B \) is given by (16), then for \( 0 < \gamma \leq 1, \) the minimizer of \( F \) is given by element-wise firm thresholding. Formally, if

\[
A^T A = \text{diag}(\alpha^2_1, \ldots, \alpha^2_N)
\]

(17)

then

\[
z_n^{\text{opt}} = \Theta \left( [A^T y] / \alpha_n^2; \lambda / \alpha_n^2, \lambda / (\gamma \alpha^2_n) \right)
\]

(18)

where \( \Theta \) stands for the firm thresholding function [58] defined entry-wise in (1).

**Definition 2** [75], [76]: Function \( f : \mathbb{R}^N \rightarrow \mathbb{R} \) is an absolutely symmetric function, if

\[
f(\pi_1, \pi_2, \ldots, \pi_N) = f([z_{\pi(1)}], [z_{\pi(2)}], \ldots, [z_{\pi(N)}])
\]

(19)

holds for any permutation \( \pi \) of \( \{1, \ldots, N\}. \)

**Proposition 1**: Let \( B^T B \) be a diagonal matrix and \( \psi_B \) be the GMC penalty function defined in (13). The subdifferential of the singular value function \( \psi_B \circ \sigma \) of a matrix \( X \) is given by the following equation:

\[
\partial(\psi_B \circ \sigma)(X) = U \text{diag}(\partial \psi_B(\sigma(X))) V^T
\]

(20)

where \( X = U \Sigma V^T \) is the SVD of \( X. \)

**Proof**: It follows from [34] that if \( B^T B \) is a diagonal matrix, the GMC penalty \( \psi_B \) is separable, comprising a sum of scalar penalties

\[
B^T B = \text{diag}(\alpha^2_1, \ldots, \alpha^2_N) \Rightarrow \psi_B(z) = \sum_{n=1}^N \phi_{\alpha_n}(z_n)
\]

(21)
where $\phi_b : \mathbb{R} \to \mathbb{R}$ is the scaled MC penalty [34, 77] defined as
$$
\phi_b(y) = \begin{cases} 
|y| - \frac{1}{2}b^2y^2, & \text{if } |y| \leq 1/b^2 \\
\frac{1}{2b^2}, & \text{otherwise.}
\end{cases}
$$
(22)

Therefore, according to Definition 2, $\psi_B$ is an absolutely symmetric function. The proof of the proposition then follows from the property of the singular value function $f \circ \sigma$ [76], where $f$ is an absolutely symmetric function.

Proposition 1 allows us to use the GMC penalty for rank approximation. We formulate the GMC penalty-regularized objective for LRSSC. Let $B \in \mathbb{R}^{N \times K}$, and let $\sigma(C)$ denote the vector of singular values of $C$. By choosing $g(C) = \psi_B(\sigma(C))$ as a rank function, and $f(C) = \psi_B(C)$ as a sparsity function in (12), we define the following nonconvex objective function:
$$
\min_{C} \frac{1}{2} \|X - XC\|_F^2 + \lambda \psi_B(\sigma(C)) + \tau \psi_B(C)
\text{s.t. } \text{diag}(C) = 0
$$
(23)

where $\psi_B$ denotes the GMC penalty defined in (13), regularized by matrix $B$. In the next section, we will show that by solving the objective (23) with ADMM, both sparsity and low-rank subproblems can be reduced to (15) with diagonal $A^T A$. In this case, the GMC penalty is closely related to the continuous exact $\ell_0$ penalty [34, 57] that approximates the convex hull of the least squares with $\ell_0$ regularization. Furthermore, diagonal $A^T A$ reduces both subproblems to the element-wise firm thresholding function, defined in (1). In the low-rank minimization subproblem, the firm thresholding operator needs to be applied to the vector of singular values.

**B. Optimization Algorithm**

To solve the optimization problem in (23), we introduce auxiliary variables $J$, $C_1$, and $C_2$ to split variables and solve subproblems independently. The reformulated objective for the GMC penalty in (23) is equivalent to
$$
\min_{J, C_1, C_2} \frac{1}{2} \|X - XC\|_F^2 + \lambda \psi_B(\sigma(C_1)) + \tau \psi_B(C_2)
\text{s.t. } J = C_1, \quad J = C_2 - \text{diag}(C_2).
$$
(24)

The augmented Lagrangian function of (24) is
$$
\mathcal{L}_{\mu_1, \mu_2}(J, C_1, C_2, A_1, A_2) = \frac{1}{2} \|X - XC\|_F^2 + \lambda \psi_B(\sigma(C_1)) + \tau \psi_B(C_2)
+ \frac{\mu_1}{2} \|J - C_1\|_F^2 + \frac{\mu_2}{2} \|J - C_2 + \text{diag}(C_2)\|_F^2
+ \langle A_1, J - C_1 \rangle + \langle A_2, J - C_2 + \text{diag}(C_2) \rangle
$$
(25)

where $\mu_1, \mu_2 > 0$ are penalty parameters and $A_1$ and $A_2$ are Lagrange multipliers.

**Update Rule for $J^{k+1}$**: Given $C_1^{k}, C_2^{k}, A_1^{k}, A_2^{k}, \mu_1^{k}, \mu_2^{k}$, we minimize the Lagrangian function in (25) with respect to $J$
$$
\min_{J} \mathcal{L}_{\mu_1^{k}, \mu_2^{k}}(C_1^{k}, C_2^{k}, J, A_1^{k}, A_2^{k}) = \min_{J} \frac{1}{2} \|X - XJ\|_F^2
+ \frac{\mu_1^{k}}{2} \|J - C_1^{k}\|_F^2 + \frac{\mu_2^{k}}{2} \|J - C_2^{k} + \text{diag}(C_2^{k})\|_F^2
+ \langle A_1^{k}, J - C_1^{k} \rangle + \langle A_2^{k}, J - C_2^{k} + \text{diag}(C_2^{k}) \rangle.
$$
(26)

The optimal solution of (26) is given by the following update:
$$
J^{k+1} = \left[ X^T X + \left( \mu_1^{k} + \mu_2^{k} \right) I \right]^{-1}
\times \left[ X^T X + \mu_1^{k} C_1^{k} + \mu_2^{k} C_2^{k} - A_1^{k} - A_2^{k} \right].
$$
(27)

**Update Rule for $C_k^{k+1}$**: Given $J^{k+1}, A_1^{k}, \mu_1^{k}$, we minimize the Lagrangian function in (25) with respect to $C_1$
$$
\min_{C_1} \mathcal{L}_{\mu_1^{k}, \mu_2^{k}}(J^{k+1}, C_1, C_2^{k}, A_1^{k}, A_2^{k})
= \min_{C_1} \lambda \psi_B(\sigma(C_1)) + \frac{\mu_1^{k}}{2} \|J^{k+1} - C_1\|_F^2
+ \|A_1^{k} - J^{k+1} - C_1\|_F^2
= \min_{C_1} \lambda \psi_B(\sigma(C_1)) + \frac{\mu_1^{k}}{2} \|J^{k+1} + A_1^{k} - C_1\|_F^2.
$$
(28)

It can be seen that (28) corresponds to the least squares problem in (15) with $A^T A = I$ and, therefore, diagonal. It follows from the condition (16) that in order to maintain convexity of the subproblem, we need to set $B = \sqrt{\mu_1^{k} \nu / \lambda I}$, $0 < \nu \leq 1$. Using Lemma 2 and Proposition 1, (28) can be solved by element-wise firm thresholding of singular values of matrix $(J^{k+1} + A_1^{k} / \mu_1^{k})$.

Specifically, let $U \Sigma V^T$ denote the SVD of matrix $(J^{k+1} + A_1^{k} / \mu_1^{k})$. The closed-form solution of (28) is given by
$$
C_1^{k+1} = U \Theta \left( \Sigma : \frac{\lambda}{\mu_1^{k}}, \frac{\lambda}{\mu_1^{k}}, \gamma \right) V^T
$$
(29)

where $\Theta$ is the firm thresholding function defined in (1).

**Update Rule for $C_2^{k+1}$**: Given $J^{k+1}, A_2^{k}, \mu_2^{k}$, we minimize the objective (25) with respect to $C_2$
$$
\min_{C_2} \mathcal{L}_{\mu_1^{k}, \mu_2^{k}}(J^{k+1}, C_1^{k+1}, C_2, A_1^{k}, A_2^{k})
= \min_{C_2} \tau \psi_B(C_2) + \frac{\mu_2^{k}}{2} \|J^{k+1} + A_2^{k} / \mu_2^{k} - C_2\|_F^2
$$
(30)

with subtraction of diagonal elements of $C_2^{k+1}$
$$
C_2^{k+1} \leftarrow C_2^{k+1} - \text{diag}(C_2^{k+1}).
$$
(31)

Similarly as the update for $C_1$, matrix $A^T A$ is the diagonal matrix and we can ensure the convexity of the subproblem (30) by setting $B = \sqrt{\mu_2^{k} \nu / \tau I}, 0 < \nu \leq 1$. Problem (30) is then solved by firm thresholding elements of matrix $(J^{k+1} + A_2^{k} / \mu_2^{k})$ and given by
$$
C_2^{k+1} = \Theta \left( J^{k+1} + A_2^{k} / \mu_2^{k} : \frac{\tau}{\mu_2^{k}}, \frac{\tau}{\mu_2^{k}}, \gamma \right)
$$
(32)

**Update Rules for Lagrange Multipliers $A_1^{k+1}, A_2^{k+1}$**: Given $J^{k+1}, C_1^{k+1}, C_2^{k+1}, \mu_1^{k}, \mu_2^{k}$, Lagrange multipliers are updated
Algorithm 1 GMC-LRSSC by ADMM Optimization

Input: Data points as columns in $X$, $[\tau, \lambda] > 0$, $0 < \gamma \leq 1$

Output: Assignment of the data points to $k$ clusters

1: Initialize: $[J, C_1, C_2, A_1, A_2] = 0$, $(\mu_i^{(0)} > 0)_{i=1}^\rho$, $\rho > 1$
2: Compute $X^TX$ for later use
3: while not converged do
4: Update $J_{k+1}$ by (27)
5: Normalize columns of $J$ to unit $\ell_2$ norm
6: Update $C_1^{k+1}$ by (29)
7: Update $C_2^{k+1}$ by (32)
8: Update $A_1^{k+1}, A_2^{k+1}$ by (33)
9: Update $\mu_i^{k+1} = \min(\rho \mu_i^k, \mu_{\text{max}})$, $i = 1, 2$
10: end while
11: Calculate affinity matrix $W = |C_1| + |C_1|^T$
12: Apply spectral clustering [9] to $W$

with the following equations:

$$A_1^{k+1} = A_1^k + \mu_1^k \left(J^{k+1} - C_1^{k+1}\right)$$
$$A_2^{k+1} = A_2^k + \mu_2^k \left(J^{k+1} - C_2^{k+1}\right).$$

Penalty parameters $\mu_1, \mu_2$ are in each step $k$ updated according to

$$\mu_i^{k+1} = \min(\rho \mu_i^k, \mu_{\text{max}}), \quad i = 1, 2$$

where $\rho > 1$ is the step size for adaptively changing $\mu_1, \mu_2$. Due to numerical reasons, $\mu_1, \mu_2$ are bounded with $\mu_{\text{max}}$, while in the convergence proof, we use formulation $\mu_i^{k+1} = \rho \mu_i^k$, $i = 1, 2$.

The main steps of the proposed algorithm are summarized in Algorithm 1.

C. Convergence Analysis

Although choosing $\gamma \in [0, 1]$ guarantees the convexity of the low-rank and sparsity subproblems and convergence of related subproblems, the objective in (23) is nonconvex. In this section, we analyze the convergence of the proposed method and show that any limit point of iteration sequence satisfies KKT conditions [78].

Proposition 2: The sequences $\{J^k, C_1^k, C_2^k, A_1^k, A_2^k\}$ generated by Algorithm 1 are all bounded.

We now state the main theorem related to the convergence property of the GMC-LRSSC algorithm.

Theorem 1: Let $Y^k = \{J^k, C_1^k, C_2^k, A_1^k, A_2^k\}_{k=1}^\infty$ be a sequence generated by Algorithm 1. Suppose that $\lim_{k \to \infty} (Y^{k+1} - Y^k) = 0$. Then, any accumulation point of the sequence $\{Y^k\}_{k=1}^\infty$ satisfies the KKT conditions for problem (24). In particular, whenever $\{Y^k\}_{k=1}^\infty$ converges, it converges to a point that satisfies KKT conditions.

The proofs of Proposition 2 and Theorem 1 are given in the Appendix.

D. Stopping Criteria and Computational Complexity

The steps in Algorithm 1 are repeated until convergence or until the maximum number of iterations is exceeded. We check the convergence by verifying the following inequalities at each iteration $k$:

$$\|J^k - C_1^k\|_\infty \leq \epsilon, \|J^k - C_2^k\|_\infty \leq \epsilon,$$

and $\|J^k - J^{k-1}\|_\infty \leq \epsilon$. We found that setting error tolerance to $\epsilon = 10^{-4}$ works well in practice. In each step, we normalize columns of matrix $J$. This normalization is frequently applied to stabilize the convergence of non-negative matrix factorization algorithms [79].

The computational complexity of Algorithm 1 is $O(n^2 + TN^2)$, where $T$ denotes the number of iterations. In the experiments, we set the maximal $T$ to 100, but on all datasets the algorithm converged within less than 15 iterations. Note that the computational complexity of the spectral clustering step is $O(N^3)$.

IV. S0/0-LRSSC ALGORITHM

A. Problem Formulation

In addition to the GMC penalty, we propose directly using $S_0$ and $\ell_0$ as constraints for low rank and sparsity. Specifically, by choosing $g(C) = ||C||_{S_0}$ as a rank function, and $f(C) = ||C||_0$ as a measure of sparsity in formulation (12), we obtain the following nonconvex optimization problem:

$$\min \frac{1}{2} \|X - XC\|_F^2 + \lambda \|C\|_{S_0} + \tau \|C\|_0$$

s.t. $\text{diag}(C) = 0$. (35)

The proximity operator $H : \mathbb{R} \to \mathbb{R}$ of $\|x\|_0$ is defined entry-wise as

$$H(y; \lambda) = \arg \min_{x \in \mathbb{R}} \left\{ \frac{1}{2}(y - x)^2 + \lambda \|x\|_0 \right\}. (36)$$

The closed-form solution of (36) at $y \in \mathbb{R}$ is the hard thresholding function defined in (2). The proximity operator of $\|C\|_{S_0}$ is the hard thresholding function applied entry-wise to the vector of singular values [60], [61].

B. Optimization Algorithm

To solve the minimization problem in (35), we split the original problem into two variables $J$ and $C$. That leads to the following objective function:

$$\min_{J, C} \frac{1}{2} \|X - XJ\|_F^2 + \lambda \|C\|_{S_0} + \tau \|C\|_0$$

s.t. $J = \text{diag}(C)$. (37)

The augmented Lagrangian function of (37) is

$$\mathcal{L}_\mu(J, C, \Lambda) \leq \frac{1}{2} \|X - XJ\|_F^2 + \lambda \|C\|_{S_0} + \tau \|C\|_0$$

$$+ \mu \|J - C + \text{diag}(C)\|_F^2$$

$$+ (\Lambda, J - C + \text{diag}(C))$$

where $\mu$ is the penalty parameter and $\Lambda$ is the Lagrange multiplier.

Update Rule for $J^{k+1}$: Given $C^k, k^k, \mu^k$, minimization of the Lagrangian function in (38) yields the following update:

$$J^{k+1} = \left[X^T X + \mu^k I\right]^{-1} \left[X^T X + \mu^k C^k - \Lambda^k\right]. (39)$$
the proximal map is given by the following equation:

$$P_{\lambda} R B = C \quad A N D \quad K O P R I V A:$$

The problem needs to be solved:

$$\lambda$$ allows us to efficiently solve problem in (40) when ized to nonconvex and nonsmooth setting in [63] and [64], \( Pf(J, C, A) \) converges regardless of the initialization, that is, regardless of the initialization, it generates a bounded sequence that has at least one limit point which is a stationary point of (38).

**Algorithm 2** \( S g / \ell_0 - L R S S C \) by ADMM Optimization

**Input:** Data points as columns in \( X, [r, \lambda] > 0, r + \lambda = 1 \)

**Output:** Assignment of the data points to \( k \) clusters

1. Initialize: \( [J, C, A] = 0, \mu^{(0)} > 0, \rho > 1 \)
2. Compute \( X^T X \) for later use
3. while not converged do
4. Update \( J^{k+1} \) by (39)
5. Normalize columns of \( J \) to unit \( \ell_2 \) norm
6. Calculate rank regularized proximal map \( P^u_g \) by (42)
7. Calculate sparsity regularized proximal map \( P^s_g \) by (44)
8. Update \( C^{k+1} = f^{0}_{g, g} \) defined in (45), (41), (43)
9. Update \( \Lambda^{k+1} \) by (46)
10. Update \( \mu^{k+1} = \min(\rho \mu^k, \mu_{\text{max}}) \)
11. end while
12. Calculate affinity matrix \( W = [C_1 + |C_1|^2 \] to \( W \)

**Proof:** The results in [63] guarantee convergence of the proximal average method. To guarantee global convergence of Algorithm 2, we rewrite the problem (37) using the following more general form:

$$\min_{C, J} f_1(C) + f_2(J)$$

subject to \( AC = BJ \)

where \( A = I, B = I, f_1(C) = \lambda \|C\|_{S_0} + \tau \|C\|_0, \) and \( f_2(J) = (1/2)\|X - XJ\|_F^2 \).

We will now show that [70, Assumptions A1–A5], which guarantee convergence in the nonconvex nonsmooth optimization problem, are satisfied. \( \| \cdot \|_0 \) and \( \| \cdot \|_{S_0} \) are non-negative lower semicontinuous functions and lower bounded. Therefore, \( f_1 \) as a sum of these functions is also lower semicontinuous and lower bounded. Furthermore, \( f_2 \) is coercive and \( B = I \), so Assumptions A1 and A4 hold. \( A = I \) and \( B = I \) imply that Assumptions A2 and A3 hold. Next, \( f_2 \) is the Lipschitz differentiable function, so Assumption A5 is also satisfied. Therefore, Assumptions A1–A5 are satisfied and Algorithm 2 converges for any sufficiently large \( \mu \) [70]. Of note, by splitting the original problem in (35) in three variables as done in GMC-LRSSC, we could not converge since Assumption A2 would not be satisfied.

Furthermore, \( \ell_0 \) and \( S_0 \) norms belong to the class of semialgebraic functions and satisfy the Kurdyka–Łojasiewicz inequality [73], [80]. The sum of the semialgebraic function is again a semialgebraic function, so \( L_\mu(J, C, A) \) in (38) is a semialgebraic function and, therefore, satisfies the Kurdyka–Łojasiewicz inequality. This allows us to establish a stronger convergence property, that is, sequence \{\( J^k, C^k, A^k \)\} generated by Algorithm 2 converges regardless of the initialization to the unique limit point [70].

**C. Convergence Analysis**

**Theorem 2:** Let \( Y^k = (J^k, C^k, A^k) \) be a sequence generated by Algorithm 2. Then, for any sufficiently large \( \mu \), Algorithm 2 converges globally.\(^1\)

\(^{1}\)That is, regardless of the initialization, it generates a bounded sequence that has at least one limit point which is a stationary point of (38).

**D. Stopping Criteria and Computational Complexity**

The steps in Algorithm 2 are repeated until convergence or when the maximum number of iterations is exceeded. The convergence is achieved when inequalities \( \|J^k - C^k\|_\infty = \epsilon \)}
and $\|J^k - J^{k-1}\|_\infty \leq \epsilon$ are satisfied. In all experiments, error tolerance $\epsilon$ is set to $10^{-4}$.

As in GMC-LRSSC, the computational complexity of Algorithm 2 is $O(nN^2 + TN^3)$, where $T$ denotes the number of iterations. We set the maximal number of iterations $T$ to 100, but the algorithm typically converged within 20 iterations.

V. EXPERIMENTAL RESULTS

In this section, we compare the clustering performance and efficiency of the proposed algorithms with state-of-the-art subspace clustering algorithms on synthetic and four real-world datasets. The performance is evaluated in terms of CE defined as

$$CE(\hat{r}, r) = \min_{\pi \in \Pi_L} \left( 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\pi(\hat{r}_i) = r_i\}} \right)$$

(48)

where $\Pi_L$ is the permutation space of $[L]$.

We compare the performance of our algorithms with state-of-the-art subspace clustering algorithms, including SSC [21]; LRR [13], [14]; closed-form low-rank subspace clustering (LRSC) [17]; SSC via orthogonal matching pursuit (SSC-OMP) [81]; thresholding-based subspace clustering (TSC) [82]; nearest subspace neighbor (NSN) [83]; LRSSC [29]; $\ell_0$-SSC ($\ell_0$-SSC) [48], [84]; and Schatten-$p$ norm minimization-based LRR [51] ($S_{2/3}$-LRR and $S_{1/2}$-LRR).

A. Experimental Setup

In all experiments, we set the parameters of GMC-LRSSC and $S_0/\ell_0$-LRSSC as follows: $\tau = 1 - \rho$, $\rho = 3$, $\mu^{\max} = 10^6$, $\in$ in stopping criteria to $10^{-4}$, and a maximum number of iterations to 100. Parameters $\lambda$ and the initial value of $\mu$ are tuned more carefully. For GMC-LRSSC, $\lambda$ is parameterized using $\alpha$ as $1/(1 + \alpha)$, where $\alpha$ is tested in range $[10^{-3}, 10^3]$ with step 10. Both $\lambda$ and $\tau$ are scaled by $\mu_0^2$. For the $S_0/\ell_0$-LRSSC parameter, $\lambda$ is optimized in the range $[0.1, 0.9]$ with step 0.1. After the best $\lambda$ is found, $\mu_0^2$ in GMC-LRSSC and $\mu_0^2$ in $S_0/\ell_0$-LRSSC are tested in the set $\{1, 3, 5, 10, 20\}$. The initial value of parameter $\mu_1$ in GMC-LRSSC is set to 0.1 in all experiments. For GMC-LRSSC, we test the nonconvexity parameter $\gamma$ in $[0.1, 0.6, 1]$. That resulted in $\gamma = 1$ on the Extended Yale B dataset, $\gamma = 0.6$ on the MNIST dataset, and $\gamma = 0.1$ on the USPS and ISOLET1 datasets. On synthetic data, we test $\gamma$ from 0.1 to 1 with step 0.1.

For other state-of-the-art algorithms, we use the source codes provided by the authors. If the best parameters are available, we set them as reported in the corresponding papers/source codes. Otherwise, we tune the parameters and retained those with the best performance. Specifically, for the SSC parameter $\alpha \in \{10, 20, 50, 80, 100, 200, 500, 800, 1000\}$, for LRR $\lambda \in \{0.05, 0.1, 0.3, 0.5, 1.2, 3.4, 5, 6, 7\}$, for LRSC $\tau \in \{0.1, 0.5, 10, 20, 50, 80, 100, 200, 500, 800\}$, and $\alpha \in \{0.1 \tau, 0.5 \tau, 0.9 \tau, 1.11 \tau, 2 \tau, 10 \tau\}$, for $S_{2/3}$-LRR and $S_{1/2}$-LRR $\lambda \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1, 1.5, 2, 3.5, 10\}$, and $\lambda$ in $\ell_0$-SSC is tuned in range $[0.1, 1]$ with step 0.1. In NSN and SSC-OMP, the number of neighbors is chosen in the set $\{2, 3, 5, 8, 10, 12, 15, 18, 20\}$. For TSC, we set $q = \max(3, \lfloor n/20 \rfloor)$. For LRSSC, we test the $\lambda$ parameter in range $[10^{-3}, 10^4]$ with step 10 on real-world datasets. In order to have completely the same setting on the synthetic data, we tune the LRSSC parameters in the same way as for GMC-LRSSC.

Parameters of all algorithms are tuned on 20 runs and for $L = \{3, 5, 10\}$ with a different random seed than in the final experiment. In the final experiment, we run each algorithm 100 times.

B. Synthetic Data

In the synthetic data experiment, we compare LRSSC, GMC-LRSSC, and $S_0/\ell_0$-LRSSC for different levels of noise and number of samples. We generate three 5-D disjoint subspaces embedded in the 100-D space. Subspace bases $\{U_i\}^3_{i=1} \in \mathbb{R}^{100 \times 5}$ are constructed such that $\text{rank}(\{U_1, U_2, U_3\}) = 10$. We randomly sample $N_i$ data points from each subspace by computing $\{X_i = U_i A_i\}^3_{i=1}$, where $\{A_i\}^3_{i=1} \in \mathbb{R}^{5 \times N_i}$ is generated from $\mathcal{N}(0, 1)$ distribution. We sample the same number of data points from each subspace, that is, $N_1 = N_2 = N_3$. We then add Gaussian noise with zero mean and vary the noise variance. Fig. 2 shows the average CE over 10 runs for a different number of samples per subspace and different noise variance.

For 50 data points per subspace and small measurement noise, $S_0/\ell_0$-LRSSC performs better than LRSSC and GMC-LRSSC. On the other hand, for larger measurement noise, GMC-LRSSC is the best performing algorithm. When we increase the number of data points to 100, GMC-LRSSC remains the best performing algorithm for most levels of noise. However, when further increasing the number of data points, $S_0/\ell_0$ performs better except for very large measurement noise. This is in line with results presented in [56] which show that $\ell_0$ quasi-norm regularization of least-squares problems outperforms $\ell_p$ regularization $0 < p \leq 1$ for small measurement noise. Whereas LRSSC and GMC-LRSSC in most cases do not improve performance when increasing the number of data points, $S_0/\ell_0$ is often able to exploit additional data.

C. Face Recognition Dataset

The Extended Yale B dataset [85], [86] consists of face images of 38 individuals (subjects). It contains 64 frontal face
images of each individual acquired under different illumination conditions. We use downsampled $48 \times 24$ pixel images and consider each vectorized image as one data point. The face images of each individual in the Yale B dataset lie approximately in a 9-D subspace [11].

We perform experiments for a different number of clusters, ranging from 5 to 30. In each experiment, we sample uniformly $L$ clusters from the total number of subjects and compute the average of CE over 100 random subsets. The results are reported in Table II with the two best results highlighted in bold. Compared to the state-of-the-art methods, GMC-LRSSC and $S_0/\ell_0$-LRSSC achieve the lowest CE on all four clustering tasks. The difference between our algorithms and the second best performing other algorithm is significant for 10, 20, and 30 clusters (FDR < 1%; Benjamini–Hochberg corrected). Importantly, by increasing the number of clusters, the difference between our $\ell_0$-based formulations and convex low-rank and sparse formulations becomes larger.

To check the effect of parameter $\gamma$ in GMC-LRSSC, we vary the parameter $\gamma$ from 0 to 1, where small $\gamma$ means that the GMC penalty $\psi_B$ is close to convex, and $\gamma = 1$ corresponds to the maximally nonconvex value of the penalty. Fig. 3 shows performance as a function of $\gamma$ values for 10, 20, and 30 clusters. On all of these tasks, larger values of $\gamma$ achieve lower CE than the smaller values.

D. Handwritten Digit Datasets

For the handwriting recognition task, we consider two datasets: 1) MNIST and 2) USPS datasets. Both datasets contain pictures of ten digits (0–9), each digit corresponding to one cluster. The MNIST dataset contains 10 000 centered $28 \times 28$ pixel images of handwritten digits. The USPS dataset consists of 92 898 handwritten digits, each measuring $16 \times 16$. The handwritten digits lie approximately in a 12-D subspace [87].

For both datasets, we use a subset of available images, sampling uniformly 50 images per digit in each run, and compute the average of CE over 100 runs. The performance comparisons for a different choice of digits are shown in Table III. On both datasets, $S_0/\ell_0$-LRSSC and GMC-LRSSC are the only algorithms that consistently achieve high performance across varying combinations of digits.

On the MNIST dataset, the $S_0/\ell_0$-LRSSC algorithm is among the best performing algorithms, and the difference increases for a larger number of clusters. GMC-LRSSC has lower performance than $S_0/\ell_0$-LRSSC for three combinations, but is still significantly better than SSC, LRR, and LRSSC. SSC and LRSSC have the best performance for digit sets $\{2, 4, 8\}$ and $\{2, 4, 6, 8, 9\}$, but they fail to give satisfactory results on other combinations of digits.

On the USPS dataset, GMC-LRSSC is slightly better than $S_0/\ell_0$-LRSSC, except for the combinations of five digits. Specifically, on the digit set $\{2, 4, 6, 8, 9\}$, $S_0/\ell_0$-LRSSC outperforms all other methods. For the hardest problems with ten clusters, GMC-LRSSC and $S_0/\ell_0$-LRSSC again have significantly better performance (FDR < 1%) than all other methods, that is, 7.2% and 4.8% higher than the second best method, respectively. Fig. 4 illustrates derived affinity matrices on the USPS dataset for ten clusters.
TABLE III
CE (%) ON MNIST AND USPS DATASETS

| Digits | Dataset  | SSC | LRR  | LRSC | SSC-Omp | TSC | NSN | LRSSC | l0-SSC | S_2/3-LRR | S_1/2-LRR | GMC-LRSSC | S_0/l0-LRSSC |
|--------|----------|-----|------|------|---------|-----|-----|-------|--------|-----------|-----------|-----------|-----------|
| 2,4,8  | MNIST    | 7.43| 14.14| 10.59| 11.06   | 12.09| 13.02| 7.01  | 7.82   | 14.80     | 15.03     | 8.66      | 8.92      |
|        | USPS     | 6.02| 10.37| 7.61 | 21.04   | 8.32 | 18.67| 7.13  | 4.88   | 8.83      | 9.70      | 4.92      | 6.40      |
| 3,6,9  | MNIST    | 3.89| 3.49 | 4.61 | 5.69    | 3.25 | 2.99 | 4.15  | 3.49   | 5.93      | 6.61      | 2.93      | 3.25      |
|        | USPS     | 2.05| 1.57 | 4.50 | 23.43   | 1.43 | 2.22 | 8.13  | 1.09   | 3.02      | 3.68      | 0.97      | 1.27      |
| 1,4,7  | MNIST    | 47.50| 45.09| 44.14| 42.21   | 33.72| 14.05| 47.09 | 45.40  | 43.42     | 42.57     | 34.50     | 27.33     |
|        | USPS     | 2.21| 4.01 | 4.33 | 58.07   | 7.45 | 9.45 | 8.61  | 3.68   | 4.27      | 5.37      | 2.65      | 3.82      |
| 2,4,6,8,9 | MNIST | 25.54| 33.54| 29.22| 29.43   | 28.95| 27.04| 26.78 | 28.94  | 34.90     | 36.29     | 27.40     | 27.20     |
|        | USPS     | 15.69| 22.30| 18.86| 53.43   | 20.35| 26.69| 18.67 | 14.31  | 19.22     | 19.95     | 15.37     | 13.38     |
| 0,1,3,5,7 | MNIST | 53.60| 33.61| 35.92| 37.51   | 30.16| 22.39| 46.86 | 33.74  | 32.50     | 33.08     | 29.80     | 27.85     |
|        | USPS     | 30.00| 22.83| 28.47| 74.66   | 25.58| 13.36| 35.44 | 30.17  | 27.41     | 27.68     | 24.76     | 10.89     |
| 0-9    | MNIST    | 47.49| 45.13| 46.88| 46.45   | 40.00| 34.81| 45.68 | 39.51  | 43.19     | 43.66     | 38.01     | 34.89     |
|        | USPS     | 28.28| 33.44| 28.58| 84.01   | 29.34| 28.43| 33.10 | 27.27  | 28.75     | 29.24     | 20.46     | 22.45     |

E. Speech Recognition Dataset

For the speech recognition task, we evaluate algorithms on the ISOLET dataset [88]. The task is to cluster subjects, where each subject spoke the name of each letter of the alphabet twice. We use dataset ISOLET1 containing 26 subjects with 30 data points from each subject. The features include spectral coefficients, contour features, sonorant, presonorant, and postsonorant features. To check whether the subspace clustering assumption holds on the ISOLET1 dataset, we compute the singular values of several subjects. Fig. 5 demonstrates that singular values decay rapidly and confirms that data points are drawn from low-dimensional subspaces.

We sample uniformly \( L = \{5, 10, 15, 20\} \) clusters from the total number of subjects over 100 random subsets. The average CEIs are reported in Table IV. For all tested numbers of clusters, \( S_0/l0-LRSSC \) is the best performing method. GMC-LRSSC is the second best method for 5 and 10 clusters, while for 15 and 20, it achieves the same result as Schatten-2/3 and Schatten-1/2 LRR.

TABLE IV
CE (%) ON THE ISOLET1 DATASET

| L     | SSC | LRR  | LRSC | SSC-Omp | TSC | NSN | LRSSC | l0-SSC | S_2/3-LRR | S_1/2-LRR | GMC-LRSSC | S_0/l0-LRSSC |
|-------|-----|------|------|---------|-----|-----|-------|--------|-----------|-----------|-----------|-----------|
| 5     | 10.98| 7.61 | 10.25| 27.79   | 11.58| 8.23| 8.63  | 9.23   | 7.45      | 8.06      | 7.07      | 6.87      |
| 10    | 17.11| 14.65| 13.54| 44.44   | 19.39| 16.04| 14.72 | 18.21  | 14.10     | 14.34     | 13.92     | 13.81     |
| 15    | 25.64| 23.14| 22.88| 34.03   | 27.07| 23.61| 23.87 | 25.73  | 20.37     | 20.24     | 20.29     | 19.90     |
| 20    | 31.05| 30.60| 27.93| 59.89   | 31.70| 27.94| 29.90 | 30.28  | 26.02     | 25.24     | 25.32     | 25.07     |

F. Computational Time and Convergence

We further test the convergence behavior of GMC-LRSSC and \( S_0/l0-LRSSC \). The convergence conditions of GMC-LRSSC are satisfied within less than 15 iterations on all four real-world datasets. Fig. 6 illustrates the convergence behavior of GMC-LRSSC on the MNIST and ISOLET1 datasets for ten clusters. \( S_0/l0-LRSSC \) converges within 2 iterations on the
The choice of norm should be decided depending on the dataset. In future work, we plan to study how different initializations affect the accuracy of the proposed nonconvex regularization-based methods. Instead of directly solving the NP-hard \( \ell_0 \) quasi-norm minimization problem, we are interested in finding a way to gradually build a solution. A possible strategy could be to start with the \( \ell_1 \)-norm solution. Since analytic formulas for thresholding function exist for \( p = 1/2 \) and \( p = 2/3 \) [89], we can gradually shrink \( p \) and use the current solution to initialize the next step of the algorithm. This approach is called the \( p \)-continuation strategy in [56].

APPENDIX

PROOF OF PROPOSITION 2 AND THEOREM 1

In this section, we first prove the boundedness of variables in Algorithm 1. This result helps us to establish the convergence property of Algorithm 1. We then prove Theorem 1 in this paper, where we show that any converging point satisfies KKT conditions [78].

**Proof of Proposition 2:** From the first-order optimality conditions of the Lagrangian function in (25), we have

\[
0 \in \partial C \psi(I_{\mu_i^1,\mu_i^2}(\mathcal{J}^{k+1}, C_1^{k+1}, C_2^{k+1}, A_1^k, A_2^k))
\]

\[
0 \in \partial C \psi(I_{\mu_i^1,\mu_i^2}(\mathcal{J}^{k+1}, C_1^{k+1}, C_2^{k+1}, A_1^k, A_2^k)).
\]  

(49)

The optimality condition of the problem in (30) implies that

\[
\partial \tau \psi_B(C_2^{k+1})_{ij} - \Lambda^{k+1}_{2,ij} = 0
\]  

(50)

where \([\partial \psi_B(C_2^{k+1})_{ij}]\) denotes the gradient of the GMC penalty \( \psi_B \) at \([C_2^{k+1}]_{ij}\). By the definition of the scaled MC penalty in (22) and using \( B = \sqrt{\mu_2^2 \gamma} I \), we have

\[
\partial \phi(c_{ij}) = \begin{cases} 
\text{sign}(c_{ij}) - \frac{\mu_2^2 \gamma}{\tau} c_{ij}, & \text{if } |c_{ij}| \leq \frac{\tau}{\mu_2^2 \gamma} \\
0, & \text{if } |c_{ij}| > \frac{\tau}{\mu_2^2 \gamma}
\end{cases}
\]  

(51)

where \(c_{ij}\) denotes \([C_2]_{ij}\).

If \(|c_{ij}| > \tau/\mu_2^2 \gamma\), then from (50) and (51), it directly follows \([A_2^{k+1}]_{ij} = 0\).

Otherwise, we obtain the following equality:

\[
[A_2^{k+1}]_{ij} = \text{sign}([C_2^{k+1}]_{ij}) - \frac{\mu_2^2 \gamma}{\tau} [C_2^{k+1}]_{ij}.
\]  

(52)

Since \(|[C_2^{k+1}]_{ij}| \leq \tau/\mu_2^2 \gamma\), it follows \(|A_2^{k+1}]_{ij}| \leq 1\). Therefore, sequence \([A_2^k]_{ij}\) is bounded.

The optimality condition of problem in (28) implies that

\[
\partial \lambda \psi_B(C_1^{k+1})_{ij} - \Lambda^{k+1}_{1,ij} = 0.
\]  

(53)

Similarly, following the proof for \(C_2\) and using Proposition 1, it can be shown that the sequence \([A_1^k]_{ij}\) is also bounded. Using the definitions of \(J^{k+1}, C_1^{k+1}\), and \(C_2^{k+1}\) as minimizers, we have the following inequalities [90]:

\[
\mathcal{L}_{\mu_i^1,\mu_i^2}(J^{k+1}, C_1^{k+1}, C_2^{k+1}, A_1^k, A_2^k) 
\)

\[
\leq \mathcal{L}_{\mu_i^1,\mu_i^2}(J^{k+1}, C_1^{k+1}, C_2^{k+1}, A_1^k, A_2^k)
\]
From the first and fourth KKT conditions, it follows:

$$\sum_{t}^{\infty} \cdots = \sum_{t}^{\infty} \cdots$$

The KKT conditions are derived as follows:

$$\sum_{t}^{\infty} \cdots$$

Therefore, sequence $\cdots$ are both bounded, $\cdots$. The boundedness of $\cdots$ and $\cdots$ is upper-bounded and $\cdots$ is also bounded. The Bolzano–Weierstrass theorem [91] then guarantees the existence of a convergent subsequence.

Proof of Theorem 1: Let $(\ddot{J}^{*}, \ddot{C}^{*}, \ddot{C}_{1}, \ddot{C}_{2}, \ddot{A}_{1}, \ddot{A}_{2})$ be a critical point of (24). The KKT conditions are derived as follows:

1. $\ddot{J}^{*} - \ddot{C}^{*} = 0$
2. $\ddot{J}^{*} - \dddot{C}^{*} = 0$
3. $-X^{T}(X - XJ^{*}) + \ddot{A}_{1}^{*} + \ddot{A}_{2}^{*} = 0$
4. $\ddot{A}_{1}^{*} \in \partial_{c_{1}} \lambda \psi B(\sigma(C_{1}^{0}))$
5. $\ddot{A}_{2}^{*} \in \partial_{c_{2}} \lambda \psi B(\sigma(C_{2}^{0}))$

From the first and fourth KKT conditions, it follows:

$$\ddot{J}^{*} + \ddot{\Lambda}_{1}^{*} \in J^{*} + \frac{\lambda}{\mu_{1}} \partial_{c_{1}} \psi B(\sigma(C_{1}^{0})) = 0$$

Let $\ddot{U}_{1}\ddot{\Sigma}_{1}V^{T}_{1}$ be the SVD of matrix $\ddot{C}_{1}^{*}$. Using Proposition 1, the right-hand side of (50) equals

$$\ddot{C}_{1}^{*} + \frac{\ddot{\lambda}}{\mu_{1}} \partial_{c_{1}} \psi B(\sigma(C_{1}^{*})) = U_{1}\ddot{\Sigma}_{1}V^{T}_{1} = U_{1}\ddot{\Sigma}_{1}^{*} + \frac{\ddot{\lambda}}{\mu_{1}} \partial_{c_{1}} \psi B(\sigma(\Sigma_{1}))V^{T}_{1}$$

$$= U_{1}\ddot{\Sigma}_{1} + \frac{\ddot{\lambda}}{\mu_{1}} \partial_{c_{1}} \psi B(\sigma(\Sigma_{1}))V^{T}_{1}$$

where $a_{1} = \mu_{1}/\lambda$ and $b_{1} = \sqrt{\mu_{2}^{*} \gamma / \lambda}$, $0 < \gamma \leq 1$. The scalar function $Q_{a,b}$ is defined as $Q_{a,b}(x) \equiv x + (1/a) \partial \phi_{b}(x)$, where $\phi_{b}$ is the scaled MC penalty defined in (22). $Q_{a,b}$ is applied element-wise to singular values of matrix $\ddot{C}_{1}^{*}$. Let $U_{2}\ddot{\Sigma}_{2}V^{T}_{2}$ be the SVD of matrix $(J^{*} + \ddot{A}_{1}^{*} \ddot{\mu})$. From (59), we obtain the following relation:

$$U_{2}\ddot{\Sigma}_{2}V^{T}_{2} = U_{2}(\sigma_{2}; \frac{\gamma}{\gamma}, \frac{\mu_{1}}{\gamma})$$

$$U_{2}\ddot{\Sigma}_{2}V^{T}_{2} = U_{2}(\sigma_{2}; \frac{\gamma}{\gamma}, \frac{\mu_{1}}{\gamma})$$

Similarly, from the second and fifth KKT conditions, we have the following relations:

$$J^{*} + \ddot{A}_{2}^{*} = J^{*} + \frac{\tau}{\mu_{2}} \partial_{c_{2}} \psi B(\ddot{C}_{2}^{0})$$

$$= C_{2}^{0} + \frac{\tau}{\mu_{2}} \partial_{c_{2}} \psi B(\ddot{C}_{2}^{0})$$

$$= C_{2}^{0} + \frac{\tau}{\mu_{2}} \partial_{c_{2}} \psi B(\ddot{C}_{2}^{0})$$

where $a_{2} = \mu_{2}^{*} / \tau$ and $b_{2} = \sqrt{\mu_{2} \gamma / \mu_{2}}$, $0 < \gamma \leq 1$. Again, applying $Q_{a\mu_{2}}^{-1}$ to both sides of (63), we obtain the following equations:

$$C_{2}^{*} = Q_{a\mu_{2}}^{-1}(J^{*} + \ddot{A}_{2}^{*})$$

$$= \Theta(\sigma_{2}; \frac{\mu_{2}}{\mu_{2}}, \frac{\gamma}{\gamma})$$

Therefore, the fourth and fifth KKT conditions can be rewritten as

$$C_{1}^{*} = U_{2}(\sigma_{1}; \frac{\gamma}{\gamma}, \frac{\mu_{1}}{\gamma})V^{T}_{1}$$

$$C_{2}^{*} = \Theta(\sigma_{2}; \frac{\mu_{2}}{\mu_{2}}, \frac{\gamma}{\gamma})$$

We now show that KKT conditions are satisfied when assumptions of Theorem 1 hold. From (33), we have

$$\ddot{A}_{1}^{k+1} - \ddot{A}_{1}^{k} = \mu_{1}^{k} \left( \ddot{J}^{k+1} - \dddot{C}_{1}^{k+1} \right)$$

$$\ddot{A}_{2}^{k+1} - \ddot{A}_{2}^{k} = \mu_{2}^{k} \left( \ddot{J}^{k+1} - \dddot{C}_{2}^{k+1} \right)$$

Since by the assumptions $(\ddot{A}_{1}^{k+1} - \ddot{A}_{1}^{k}) \rightarrow 0$ and $(\ddot{A}_{2}^{k+1} - \ddot{A}_{2}^{k}) \rightarrow 0$, then the first two KKT conditions are satisfied.
From the first two conditions, it follows that when \( J^{k+1} - J^k \rightarrow 0 \), the third KKT condition is satisfied.

Next, using the update for \( C_1 \) in (29), we obtain the following equation:

\[
C_1^{k+1} - C_1^k = U_2 \Theta \left( \Sigma + \frac{\lambda}{\mu_1^2} - \frac{\lambda}{\gamma \mu_1^2} \right) V_2^T - C_1^k
\]

(68)

where \( U_2 \Sigma V_2^T \) is the SVD of matrix \( (J^{k+1} + A_1^k/\mu_1^2) \).

From the update rule for \( C_2 \) in (32), it follows:

\[
c_2^{k+1} - c_2^k = \Theta \left( J^{k+1} + A_1^k + \frac{\tau}{\mu_2^2} - \frac{\tau}{\gamma \mu_2^2} \right) - C_2^k.
\]

(69)

When \( c_1^{k+1} - c_1^k \rightarrow 0 \) and \( c_2^{k+1} - c_2^k \rightarrow 0 \), follow the equations in (65). Since \( \{Y_k\}_{k=1}^{\infty} \) is bounded and (66)–(69) go to zero, we conclude that the sequence \( \{Y_k\}_{k=1}^{\infty} \) asymptotically satisfies the KKT conditions in (58).

\[\text{ACKNOWLEDGMENT}\]

The authors would like to thank S. Matsushima for his helpful comments on improving the statements of Proposition 2 and Theorem 1. The authors would also like to thank the Editor and reviewers for their comments which helped to improve this paper.

\[\text{REFERENCES}\]

[1] X. Wang and X. Tang, “A unified framework for subspace face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, pp. 1222–1228, Sep. 2004.
[2] 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. 2009.
[3] S. Rao, R. Tron, R. Vidal, and Y. Ma, “Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 10, pp. 1832–1845, Oct. 2010.
[4] F. de la Torre, M. Borgeaud, V. Gass, J.-P. Thiran, and D. Tuia, “Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 6, pp. 3410–3420, Jun. 2016.
[5] H. Song, W. Yang, N. Zhong, and X. Xu, “Unsupervised classification of PolSAR imagery via kernel sparse subspace clustering,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 10, pp. 1487–1491, Oct. 2016.
[6] A. W.-C. Liew, N.-F. Law, and H. Yan, “Missing value imputation for gene expression data: Computational techniques to recover missing data from available information,” Brief Bioinform., vol. 12, no. 5, pp. 498–513, 2011.
[7] R. Vidal, “Subspace clustering,” IEEE Signal Process. Mag., vol. 28, no. 2, pp. 52–68, Mar. 2011.
[8] S. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 888–905, Aug. 2000.
[9] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Proc. Adv. Neural Inf. Process. Syst., 2001, pp. 849–856.
[10] U. von Luxburg, “A tutorial on spectral clustering,” Stat. Comput., vol. 17, no. 4, pp. 395–416, 2007.
[11] 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.
[12] X. Peng, Z. Yu, Z. Yi, and H. Tang, “Constructing the L2-graph for robust subspace learning and subspace clustering,” IEEE Trans. Cybern., vol. 47, no. 4, pp. 1053–1066, Apr. 2017.
[13] G. Liu, Y. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in Proc. 26th Int. Conf. Mach. Learn., 2010, pp. 663–670.
Maria Brbić received the graduation degree with the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, where she is currently pursuing the Ph.D. degree in computer science. She is with Rudjer Bošković Institute, Zagreb. Her current research interest includes unsupervised learning with applications to computational biology.

Ivica Kopriva (M’98–SM’04) received the Ph.D. degree in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 1998. He was a Senior Research Scientist with the ECE Department, George Washington University, Washington, DC, USA, from 2001 to 2005. Since 2006, he has been a Senior Scientist with Rudjer Bošković Institute, Zagreb. His current research interests include algorithms for blind source separation and unsupervised learning with applications in medical imaging, spectroscopy, and variable selection.

Dr. Kopriva was a recipient of the 2008 State Award for Science of the Republic of Croatia.