Sleem *et al.*
EURASIP Journal on Advances in Signal Processing (2024) 2024:22
https://doi.org/10.1186/s13634-024-01114-6

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Lp quasi-norm minimization: algorithm and applications

Omar M. Sleem[1*], M. E. Ashour[2], N. S. Aybat[3] and Constantino M. Lagoa[1]

*Correspondence: oms46@psu.edu

[1] Department of Electrical Engineering, Pennsylvania State University, State College, PA 16802, USA
[2] Wireless R&D Department, Qualcomm Technologies, Inc, San Diego, CA 92121, USA
[3] Department of Industrial and Manufacturing Engineering, Pennsylvania State University, State College, PA 16802, USA

Abstract

Sparsity finds applications in diverse areas such as statistics, machine learning, and signal processing. Computations over sparse structures are less complex compared to their dense counterparts and need less storage. This paper proposes a heuristic method for retrieving sparse approximate solutions of optimization problems via minimizing the ℓ_p quasi-norm, where $0 < p < 1$. An iterative two-block algorithm for minimizing the ℓ_p quasi-norm subject to convex constraints is proposed. The proposed algorithm requires solving for the roots of a scalar degree polynomial as opposed to applying a soft thresholding operator in the case of ℓ_1 norm minimization. The algorithm's merit relies on its ability to solve the ℓ_p quasi-norm minimization subject to any convex constraints set. For the specific case of constraints defined by differentiable functions with Lipschitz continuous gradient, a second, faster algorithm is proposed. Using a proximal gradient step, we mitigate the convex projection step and hence enhance the algorithm's speed while proving its convergence. We present various applications where the proposed algorithm excels, namely, sparse signal reconstruction, system identification, and matrix completion. The results demonstrate the significant gains obtained by the proposed algorithm compared to other ℓ_p quasi-norm based methods presented in previous literature.

Keywords: Sparsity, Compressed sensing, Rank minimization, Alternating direction method of multipliers, System identification, Matrix completion, Proximal gradient method

1 Introduction

1.1 Motivation

In numerical analysis and scientific computing, a sparse matrix/array is the one with many of its elements being zeros. The number of zeros divided by the total number of elements is called sparsity. Sparse data is often easier to store and process. Hence, techniques for deriving sparse solutions and exploiting them have attracted the attention of many researchers in various engineering fields like machine learning, signal processing, and control theory.

The taxonomy of sparsity can be studied through the Rank Minimization Problem (RMP). It has been lately considered in many engineering applications including control

design and system identification. This is because the notions of complexity and system order can be closely related to the matrix rank. The RMP can be formulated as follows:

$$\min_{\mathbf{X}\in\mathcal{M}} \quad \mathbf{Rank}(\mathbf{X}), \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathcal{M} \subset \mathbb{R}^{m\times n}$ is a convex set. The problem (1) in its generality is NP-hard [1]. Therefore, polynomial time algorithms for solving large-scale problems of the form in (1) are not currently known. Hence, recently adopted methods for solving such problems are approximate and structured heuristics. A special case of RMP is the Sparse Vector Recovery (SVR) problem involving ℓ_0 pseudo-norm minimization given by:

$$\min_{\mathbf{x}\in\mathcal{V}} \quad \|\mathbf{x}\|_0, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{V} \subset \mathbb{R}^n$ is a closed convex set and $\|\cdot\|_0$ counts the number of the non-zero elements of its argument. From the definition of the rank being the number of non-zero singular values of a matrix, it can be easily realized that (1) is a generalized form of (2).

Numerous studies, which will be expounded upon in the subsequent section, have individually addressed effective solution methods for the problems presented in (1) and (2). These approaches utilize Schatten-p and ℓ_p quasi-norm relaxations, respectively. However, existing methods in this domain often either assume a predefined structure for the convex set $\mathcal{M}$ in (1) or exclusively cater to the specialized case articulated in (2). Consequently, these methods lack comprehensive applicability. Leveraging the inherent relationship between the Schatten-p quasi-norm and the ℓ_p quasi-norm of matrix singular values, we endeavor to formulate an efficient heuristic method based on Schatten-p relaxation. This method is devised to address both problems in a unified manner. The proposed approach begins with the introduction of an algorithm for solving the ℓ_p quasi-norm relaxation of the SVR problem presented in (2). Subsequently, recognizing that (2) constitutes a specific case of (1), we utilize the developed ℓ_p quasi-norm minimization algorithm as a foundational component for constructing the envisaged generalized algorithm for RMPs.

1.2 Related work

1.2.1 Sparse vector recovery

Given that many signals exhibit sparsity or compressibility, the SVR problem has found widespread applications in fields such as object recognition, classification, and compressed sensing, as evidenced by studies such as [2–4]. The concept of sparse representation of signals and systems has been extensively discussed in [5], where the authors conducted a comprehensive review of both theoretical and empirical results pertaining to sparse optimization. They also derived the sufficient conditions necessary for ensuring uniqueness, stability, and computational feasibility. Moreover, [5] explores diverse applications of the SVR problem, contending that in certain tasks involving denoising and compression, methods rooted in sparse optimization offer state-of-the-art solutions.

The problem of constructing sparse solutions for undetermined linear systems has garnered significant attention. A survey conducted in [6] comprehensively examined existing algorithms for sparse approximation. The reviewed methods encompassed various approaches, including greedy methods [7, 8], techniques rooted in convex relaxation [3,

4], those employing non-convex optimization strategies [9, 10], and approaches necessitating brute force [11]. The authors discussed the computational demands of these algorithms and elucidated their interrelationships.

Sparse optimization problems of the form $\min f(\mathbf{x}) + \mu g(\mathbf{x})$ have been extensively explored in the literature, where $g(\mathbf{x})$ serves as a sparsity-inducing function, f represents a loss function capturing measurement errors, and $\mu > 0$ functions as a trade-off parameter balancing data fidelity and sparsity. In [12], the authors addressed a sparse recovery problem involving a set of corrupted measurements. By defining $g(\cdot)$ as the ℓ_1 norm, they established a sufficient condition for exact sparse signal recovery, specifically the Restricted Isometry Property (RIP).

Motivated by the convergence of the ℓ_p quasi-norm to the ℓ_0 pseudo-norm as $p \to 0$, the problem was extended in [13] by setting g as the ℓ_p quasi-norm for $p \in (0, 1)$. The authors presented theoretical results showcasing the ℓ_p quasi-norm's capability to recover sparse signals from noisy measurements. Under more relaxed RIP conditions, it was demonstrated that the ℓ_p quasi-norm provides superior theoretical guarantees in terms of stability and robustness compared to ℓ_1 minimization.

In [9], the authors considered the problem of SVR via ℓ_p quasi-norm minimization from a limited number of linear measurements of the target signal. However, the proposed approach faced limitations due to its higher computational complexity compared to the ℓ_1 norm. In [14], Fourier-based algorithms for convex optimization were leveraged to solve sparse signal reconstruction problems via ℓ_p quasi-norm minimization, demonstrating a combination of the construction capabilities of non-convex methods with the speed of convex ones.

An alternative approach for sparse reconstruction was proposed in [15], replacing the non-convex function with a quadratic convex one. Furthermore, [16] introduced an Alternating Direction Method of Multipliers (ADMM) [17] based algorithm enforcing both sparsity and group sparsity using non-convex regularization. Additionally, [18] proposed an iterative half-thresholding algorithm for expedited solutions of $\ell_{0.5}$ regularization. The authors not only established the existence of the resolvent of the gradient of the $\ell_{0.5}$ quasi-norm but also derived its analytic expression and provided a thresholding representation for the solutions. The convergence of this iterative half-thresholding algorithm was studied in [19], demonstrating its convergence to a local minimizer of the regularized problem with a linear convergence rate.

Conditions for the convergence of an ADMM algorithm aimed at minimizing the sum of a smooth function with a bounded Hessian and a non-smooth function are established in [20]. In [21], the convergence of ADMM is analyzed for the minimization of a non-convex and potentially non-smooth objective function subject to equality constraints. The derived convergence guarantee extends to various non-convex objectives, encompassing piece-wise linear functions, ℓ_p quasi-norm, and Schatten-p quasi-norm ($0 < p < 1$), while accommodating non-convex constraints. Several works have explored the ℓ_{1-2} relaxation objective, defined the difference between ℓ_1 and ℓ_2 norms, i.e., $\ell_1 - \ell_2$, with [22] providing a theoretical analysis on SVR through weighted ℓ_{1-2} minimization when partial support information is available. Recovery conditions for exact SVR within a ℓ_{1-2} objective framework are derived in [23, 24], along with references therein, establishing the theoretical foundation for ensuring accurate SVR outcomes.

1.2.2 Rank minimization

In [25], the authors sought to determine the least order dynamic output feedback, utilizing the formulation akin to (1), capable of stabilizing a linear time-invariant system. Their approach involved minimizing the trace, as opposed to the rank, resulting in a Semi-Definite Program (SDP) amenable to efficient solution techniques. Notably, their solution was specifically applicable to symmetric and square matrices. Building upon this work, [26] introduced a generalization of the aforementioned approach. This extension involved replacing the rank in the objective function with the summation of the singular values of the matrix, commonly known as the nuclear norm. The authors demonstrated that this modification yields the convex envelope of the non-convex rank objective, reducing to the original trace heuristic when the decision matrix assumes the form of a symmetric Positive Semi-Definite (PSD) matrix.

In [27], an alternative heuristic based on the logarithm of the determinant was introduced as a surrogate for rank minimization within the subspace of PSD matrices. The authors demonstrated that this formulation could be effectively solved through a sequence of trace minimization problems. In a related study, [28] delved into existing trace and log determinant heuristics, exploring their applications for computing a low-rank approximation in various scenarios. Specifically, the applications encompassed obtaining simple data models with interpretability by approximating *covariance matrices* for a given dataset.

Drawing inspiration from the success of the ℓ_p quasi-norm ($0 < p < 1$) for sparse signal reconstruction, an alternative method aims to enforce low-rank structure using the Schatten-p quasi-norm. This norm is defined as the ℓ_p quasi-norm of the singular values. In [29], the authors addressed the matrix completion problem, which involves constructing a low-rank matrix based on a subset of its entries. Instead of minimizing the nuclear norm, they proposed a Schatten-p quasi-norm formulation and investigated its convergence properties. To enhance the robustness of the solution, [30] combined the Schatten-p quasi-norm for low-rank recovery with the ℓ_p quasi-norm ($0 < p \leq 1$) of prediction errors on the observed entries. The authors introduced an algorithm based on ADMM, which demonstrated superior numerical performance compared to other completion methods. In a non-convex approach for matrix optimization problems involving sparsity, [31] developed a technique using a generalized shrinkage operation. This method enhances the separation of moving objects from the stationary background by decomposing video into low-rank and sparse components, presenting advantages over the convex case.

1.3 Contributions

In spite of the commendable performance exhibited by the array of algorithms outlined in Sects. 1.2.1 and 1.2.2, each designed to address different relaxations of (1) and (2), it is essential to acknowledge their problem-specific nature, primarily grounded in the specific structural attributes of the convex constraint sets they address. This issue of specialization results in a lack of generality across problem domains.

In this paper, we present a versatile algorithm grounded in the principles of projections onto constraint sets. A distinctive feature of this approach lies in its minimal

reliance on problem-specific structural constraints, prioritizing the foundational characteristic of closed convexity. The works [32, 33] delve into a comprehensive exploration, analyzing the intrinsic attributes of the projection operation onto constraint sets. While the former addresses the issue without incorporating a crucial coupling condition for polynomial equations, the latter assumes prior knowledge of the projection technique for each given point on ℓ_p balls.

Initially, we propose an ADMM based algorithm, termed as ℓ_p Quasi-Norm ADMM (pQN-ADMM), designed to solve the ℓ_p quasi-norm relaxation of (2). At each iteration, the pivotal operation involves computing Euclidean projections onto specific convex and non-convex sets. Notably, the algorithm exhibits two key properties: 1) Its computational complexity aligns with that of ℓ_1 minimization algorithms, with the additional task of solving for the roots of a polynomial; 2) It does not necessitate a specific structure for the convex set.

Subsequently, we extend the application of the proposed algorithm to address the relaxation of (1) by embracing the Schatten-p quasi-norm. In this extension, we leverage the equivalence between minimizing the ℓ_p quasi-norm of the vector of singular values and minimizing the Schatten-p quasi-norm. Our study encompasses the following numerical instances:

1 An example employing SVR, wherein the primary objective is the recovery of the sparsest feasible vector from given realizations.
2 A matrix completion example, where the overarching goal is the reconstruction of an unknown low-rank matrix based on a limited subset of observed entries.
3 Addressing a time-domain system identification problem, specifically tailored for minimum-order system detection.

Our numerical results compellingly showcase the competitiveness of pQN-ADMM when bench-marked against several state-of-the-art baseline methods.

Conclusively, given the inherent reliance of the derived algorithm on a convex projection step in each iteration, our endeavor is directed towards the formulation of an expedited algorithm accompanied by a rigorous mathematical convergence guarantee. Focusing on a subset of problems where the constraint set manifests as a polytope, we leverage principles from the Proximal Gradient (PG) method to formulate a rapid algorithm. The convergence of this algorithm is established with a rate of $O(\frac{1}{K})$, where K denotes the iteration budget assigned to the algorithm.

2 Notation

Unless otherwise specified, we denote vectors with lowercase boldface letters, i.e., $\mathbf{x}$, with i-th entry as x_i, while matrices are in uppercase, i.e. $\mathbf{X}$, with (i, j)-th entry as $x_{i,j}$. For an integer $n \in \mathbb{Z}_+$, $[n] \triangleq \{1, \ldots, n\}$. $\mathbf{1}$ represents a vector of all entries equal to 1, while $\mathbb{1}_{\mathcal{G}}(.)$ is an indicator function to the set $\mathcal{G}$, i.e., it evaluates to zero if its argument belongs to the set $\mathcal{G}$ and is $+\infty$ otherwise.

For a vector $\mathbf{x} \in \mathbb{R}^n$, the general ℓ_p norm is defined as:

$$\|\mathbf{x}\|_p \triangleq \left(\sum_{i\in[n]} |x_i|^p \right)^{\frac{1}{p}}, \tag{3}$$

where, we let $\|\mathbf{x}\|$ be the well-known Euclidean norm, i.e., $p = 2$. When $0 < p < 1$, the expression in (3) is termed as a quasi-norm satisfying the same axioms of the norm except the triangular inequality making it a non-convex function.

For a matrix $\mathbf{X}$, $\|\mathbf{X}\|$ represents the spectral norm, which is defined as the square root of the maximum eigenvalue of the matrix $\mathbf{X}^{\mathrm{H}}\mathbf{X}$. $\mathbf{X}^{\mathrm{H}}$ refers to the complex conjugate transpose of $\mathbf{X}$, denoted as $\mathbf{X}^{\top}$. On the other hand, $\|.\|_{\mathrm{f}}$ signifies the Frobenius norm of a matrix.

The Schatten-p quasi-norm of a matrix $\mathbf{X}$ is defined as:

$$\|\mathbf{X}\|_{p,*} \triangleq \left(\sum_{i\in[\min\{m,n\}]} \sigma_i(\mathbf{X})^p \right)^{\frac{1}{p}}, \tag{4}$$

where $\sigma_i(\mathbf{X})$ is the i-th singular value of the matrix $\mathbf{X}$. We utilize the $*$ subscript in (4) to differentiate the matrix Schatten-p quasi-norm from vector ℓ_p case defined in (3). When $p = 1$, (4) yields the nuclear norm which is the convex envelope of the rank function. Throughout the paper, we consider a non-convex relaxation for the rank function, specifically $p = 1/2$.

We define the ceiling operator, denoted as $\lceil\cdot\rceil$, the vectorization operator $\mathrm{vec}(\mathbf{X}) \in \mathbb{R}^{\mathrm{mn}}$, representing the vector obtained by stacking the columns of the matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, and the Hankel operator *Hankel*(.), producing a Hankel matrix from the provided vector arguments. We define the sign operator, denoted as $\mathrm{sign}(\cdot)$, which outputs -1, 0, or 1 corresponding to a negative, zero, or positive argument, respectively.

3 Sparse vector recovery algorithm

3.1 Problem formulation

This section develops a method for approximating the solution of (2) using the following relaxation:

$$\min_{\mathbf{x}\in\mathcal{V}} \quad \|\mathbf{x}\|_p^p, \tag{5}$$

where $p \in (0, 1]$ and $\mathcal{V}$ is a closed convex set. Problem (5) is convex for $p \geq 1$; hence, can be solved to optimality efficiently. However, the problem becomes non-convex when $p < 1$. We present a gradient-based algorithm and consequently, it may not always converge to a global optimum solution but only to a stationary point. An epigraph equivalent formulation of (5) is obtained by introducing the variable $\mathbf{t} = [t_i]_{i\in[n]}$:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{t}} \quad \mathbf{1}^{\top}\mathbf{t}, \\ &\text{s.t.} \quad t_i \geq |x_i|^p, \quad i \in [n], \quad \mathbf{x} \in \mathcal{V}. \end{aligned} \tag{6}$$

Let $\mathcal{X} \subset \mathbb{R}^2$ denote the epigraph of the scalar function $|x|^p$, i.e., $\mathcal{X} = \{(x, t) \in \mathbb{R}^2 : t \geq |x|^p\}$, which is a non-convex set for $p < 1$. Then, (6) can be cast as:

$$\min_{\mathbf{x},\mathbf{t}} \sum_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbf{1}^\top \mathbf{t}, \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{V}. \tag{7}$$

ADMM, as introduced in [17], leverages the inherent problem structure to partition the optimization process into simpler sub-problems, which are solved iteratively. To achieve this, auxiliary variables $\mathbf{y} = [y_i]_{i\in[n]}$ and $\mathbf{z} = [z_i]_{i\in[n]}$ are introduced, leading to an ADMM reformulation of the problem defined in (7):

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{t},\mathbf{y},\mathbf{z}} \quad \sum_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbb{1}_{\mathcal{V}}(\mathbf{y}) + \mathbf{1}^\top \mathbf{z}, \\ &\text{s.t.} \quad \mathbf{x} = \mathbf{y} : \boldsymbol{\lambda}, \quad \mathbf{t} = \mathbf{z} : \boldsymbol{\theta}. \end{aligned} \tag{8}$$

The dual variables associated with the constraints $\mathbf{x} = \mathbf{y}$ and $\mathbf{t} = \mathbf{z}$ are $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$, respectively. Throughout the paper, the colons in the constraints of an optimization problem serve as a means to associate the constraint (appearing on the left side of the colon) with its corresponding Lagrange multiplier (found on the right side of the colon). The Lagrangian function corresponding to (8) augmented with a quadratic penalty on the violation of the equality constraints with penalty parameter $\rho > 0$, is given by:

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x},\mathbf{t},\mathbf{y},\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\theta}) &= \sum_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbb{1}_{\mathcal{V}}(\mathbf{y}) + \mathbf{1}^\top \mathbf{z} \\ &+ \boldsymbol{\lambda}^\top(\mathbf{x}-\mathbf{y}) + \boldsymbol{\theta}^\top(\mathbf{t}-\mathbf{z}) + \frac{\rho}{2}\Big(\|\mathbf{x}-\mathbf{y}\|^2 + \|\mathbf{t}-\mathbf{z}\|^2\Big). \end{aligned} \tag{9}$$

Considering the two block variables $(\mathbf{x},\mathbf{t})$ and $(\mathbf{y},\mathbf{z})$, ADMM consists of the following iterations:

$$(\mathbf{x},\mathbf{t})^{k+1} = \operatorname*{argmin}_{\mathbf{x},\mathbf{t}} \mathcal{L}_\rho(\mathbf{x},\mathbf{t},\mathbf{y}^k,\mathbf{z}^k,\boldsymbol{\lambda}^k,\boldsymbol{\theta}^k), \tag{10}$$

$$(\mathbf{y},\mathbf{z})^{k+1} = \operatorname*{argmin}_{\mathbf{y},\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^{k+1},\mathbf{t}^{k+1},\mathbf{y},\mathbf{z},\boldsymbol{\lambda}^k,\boldsymbol{\theta}^k), \tag{11}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}), \tag{12}$$

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \rho(\mathbf{t}^{k+1} - \mathbf{z}^{k+1}). \tag{13}$$

Given the augmented Lagrangian function expressed in (9), it is evident from (10) that the variables $\mathbf{x}$ and $\mathbf{t}$ are iteratively updated by solving the following non-convex problem:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{t}} \quad \|\mathbf{x} - \mathbf{y}^k + \frac{\boldsymbol{\lambda}^k}{\rho}\|^2 + \|\mathbf{t} - \mathbf{z}^k + \frac{\boldsymbol{\theta}^k}{\rho}\|^2, \\ &\text{s.t.} \quad (x_i,t_i) \in \mathcal{X}, \quad i \in [n]. \end{aligned} \tag{14}$$

Exploiting the separable structure of (14), one immediately concludes that (14) can be split into n independent 2-dimensional problems that can be solved in parallel, i.e., for each $i \in [n]$:

$$(x_i, t_i)^{k+1} = \Pi_{\mathcal{X}} \left(y_i^k - \frac{\lambda_i^k}{\rho}, z_i^k - \frac{\theta_i^k}{\rho} \right), \tag{15}$$

where $\Pi_{\mathcal{X}}(.)$ denotes the Euclidean projection operator onto the set $\mathcal{X}$. Furthermore, (9) and (11) imply that $\mathbf{y}$ and $\mathbf{z}$ are independently updated as follows:

$$\mathbf{y}^{k+1} = \Pi_{\mathcal{Y}} \left(\mathbf{x}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho} \right), \tag{16}$$

$$\mathbf{z}^{k+1} = \mathbf{t}^{k+1} + \frac{\boldsymbol{\theta}^k - \mathbf{1}}{\rho}. \tag{17}$$

Algorithm 1 ADMM ($\rho > 0$)

Algorithm 1 ADMM ($\rho > 0$)

Initialize: $\mathbf{y}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$, $\boldsymbol{\theta}^0$
for $k \geq 0$ **do**
  $(x_i, t_i)^{k+1} \leftarrow \Pi_{\mathcal{X}} \left(y_i^k - \frac{\lambda_i^k}{\rho}, z_i^k - \frac{\theta_i^k}{\rho} \right), \forall i \in [n]$
  $\mathbf{y}^{k+1} \leftarrow \Pi_{\mathcal{Y}} \left(\mathbf{x}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho} \right)$
  $\mathbf{z}^{k+1} \leftarrow \mathbf{t}^{k+1} + \frac{\boldsymbol{\theta}^k - \mathbf{1}}{\rho}$
  $\boldsymbol{\lambda}^{k+1} \leftarrow \boldsymbol{\lambda}^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$
  $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k + \rho(\mathbf{t}^{k+1} - \mathbf{z}^{k+1})$.

Algorithm 1 summarizes the proposed ADMM algorithm. It is clear that $\mathbf{z}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\theta}$ merit closed-form updates. However, updating $(\mathbf{x}, \mathbf{t})$ requires solving n non-convex problems. Our strategy for dealing with this issue is presented in the following section.

3.2 Non-convex projection

In this section, we present the method used to tackle the non-convex projection problem required to update $\mathbf{x}$ and $\mathbf{t}$.

As it is clear from (15), $\mathbf{x}$ and $\mathbf{t}$ can be updated element-wise via performing a projection operation onto the non-convex set $\mathcal{X}$, one for each $i \in [n]$. The n projection problems can be run independently in parallel. We now outline the proposed idea for solving one such projection, i.e., we suppress the dependence on the index of the entry of $\mathbf{x}$ and $\mathbf{t}$. For $(\bar{x}, \bar{t}) \in \mathbb{R}^2$, $\Pi_{\mathcal{X}}(\bar{x}, \bar{t})$ entails solving:

$$\min_{x,t} \quad g(x,t) \triangleq (t - \bar{t})^2 + (x - \bar{x})^2, \quad \text{s.t.} \quad t \geq |x|^p. \tag{18}$$

If $\bar{t} \geq |\bar{x}|^p$, then trivially $\Pi_{\mathcal{X}}(\bar{x}, \bar{t}) = (\bar{x}, \bar{t})$. Thus, we focus on the case in which $\bar{t} < |\bar{x}|^p$. The following theorem states the necessary optimality conditions for (18).

Theorem 1 *Let $\bar{t} < |\bar{x}|^p$, and (x^*, t^*) be an optimal solution of (18). Then, the following properties are satisfied*:

(*a*) $\text{sign}(x*) = \text{sign}(\bar{x})$,
(*b*) $t^* \geq \bar{t}$,
(*c*) $|x^*|^p \geq \bar{t}$,
(*d*) $t^* = |x^*|^p$.

Proof We prove the statements by contradiction as follows:

(a) Suppose that $\text{sign}(x^*) \neq \text{sign}(\bar{x})$, then:

$$|x^* - \bar{x}| = |x^* - 0| + |\bar{x} - 0| > |\bar{x} - 0|, \tag{19}$$

i.e., $(x^* - \bar{x})^2 > (0 - \bar{x})^2$. Hence, $g(x^*, t^*) - g(0, t^*) > 0$. Moreover, the feasibility of (x^*, t^*) implies that $t^* > 0$. Thus, $(0, t^*)$ is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(b) Assume that $t^* < \bar{t}$. Then:

$$g(x^*, t^*) - g(x^*, \bar{t}) = (t^* - \bar{t})^2 > 0. \tag{20}$$

Furthermore, by the feasibility of (x^*, t^*), we have $|x^*|^p \leq t^* < \bar{t}$. Thus, $(x^*, \bar{t})$ is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(c) Suppose that $|x^*|^p < \bar{t}$, i.e.,

$$-\bar{t}^{\frac{1}{p}} < x^* < \bar{t}^{\frac{1}{p}}. \tag{21}$$

We now consider two cases, $\bar{x} > 0$ and $\bar{x} < 0$. First, let $\bar{x} > 0$. Then, we have by (*a*) and (21) that $0 < x^* < \bar{t}^{\frac{1}{p}}$. Since $\bar{t} < |\bar{x}|^p$, i.e., $(\bar{x}, \bar{t}) \notin \mathcal{X}$, therefore $\bar{t}^{\frac{1}{p}} < \bar{x}$ and hence, $0 < x^* < \bar{t}^{\frac{1}{p}} < \bar{x}$. Pick $x_0 > 0$ such that $|x_0|^p = \bar{t}$, i.e., $x_0 = \bar{t}^{\frac{1}{p}}$. Then clearly, $x^* < x_0 < \bar{x}$. Thus, we have:

$$g(x^*, t^*) - g(x_0, t^*) = (x^* - \bar{x})^2 - (x_0 - \bar{x})^2 > 0, \tag{22}$$

where the last inequality follows the just proven identity that $x^* < x_0 < \bar{x}$. Moreover, we have $|x_0|^p = \bar{t} \leq t^*$ by (*b*). Thus, (x_0, t^*) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*). On the other hand, let $\bar{x} < 0$. Then, we have by (*a*) and (21) that $-\bar{t}^{\frac{1}{p}} < x^* < 0$. Since $\bar{t} < |\bar{x}|^p$, i.e., $(\bar{x}, \bar{t}) \notin \mathcal{X}$, then $\bar{t}^{\frac{1}{p}} < |\bar{x}|$, i.e., $\bar{x} < -\bar{t}^{\frac{1}{p}}$. Therefore, $\bar{x} < -\bar{t}^{\frac{1}{p}} < x^*$. Pick $x_0 < 0$ such that $|x_0|^p = \bar{t}$, i.e., $x_0 = -\bar{t}^{\frac{1}{p}}$. Then, (22) also holds when $\bar{x} < 0$. Note that $|x_0|^p = \bar{t} \leq t^*$ by (*b*). Thus, (x_0, t^*) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(d) The feasibility of (x^*, t^*) eliminates the possibility that $t^* < |x^*|^p$. Now let $t^* > |x^*|^p$ and pick $t_0 = |x^*|^p$. Then, $\bar{t} \leq |x^*|^p = t_0 < t^*$, where the first inequality follows from (*c*). Then, $0 \leq t_0 - \bar{t} < t^* - \bar{t}$. Thus, we have:

$$g(x^*, t^*) - g(x^*, t_0) = (t^* - \bar{t})^2 - (t_0 - \bar{t})^2 > 0, \tag{23}$$

Furthermore, the feasibility of (x^*, t_0) follows trivially from the choice of t_0. Thus, (x^*, t_0) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

This concludes the proof. □

We now make use of the fact that for (18), an optimal solution (x^*, t^*) satisfies $t^* = |x^*|^p$ and hence, (18) reduces to solving:

$$\min_{x} \quad (|x|^p - \bar{t})^2 + (x - \bar{x})^2. \tag{24}$$

The first order necessary optimality condition for (24) implies the following:

$$p|x^*|^{p-1}\text{sign}(x^*)(|x^*|^p - \bar{t}) + x^* - \bar{x} = 0. \tag{25}$$

By the symmetry of the function $|x|^p$, without loss of generality, assume that $x^* > 0$ and let $0 < p = \frac{s}{q} < 1$ for some $s, q \in \mathbb{Z}_+$. A change of variables $a^q = x^*$ plugged in (25) shows that finding an optimal solution for (18) reduces to finding a root of the following scalar degree $2q$ polynomial:

$$a^{2q} + \frac{s}{q}\left(a^{2s} - \bar{t}a^s\right) - \bar{x}a^q. \tag{26}$$

To determine $\Pi_{\mathcal{X}}(\bar{x}, \bar{t})$, the objective is to find a root denoted as a^* for the polynomial in (26), while ensuring that the pair (a^{*^q}, a^{*^s}) minimizes the function $g(x, t)$. Algorithm 2 provides a summary of the method employed to address problem (18).

When both $x^* = 0$ and $t^* = 0$, the objective function evaluates to $g(0, 0) = \bar{x}^2 + \bar{t}^2$. To optimize the objective function while upholding the constraint $t \geq |x|^p$, the choice is made to set $x^* = 0$ and $t^* = \max\{0, \bar{t}\}$. This decision ensures that $g(0, t^*) \leq g(0, 0)$. In instances where $\bar{x} = 0$ and $\bar{t} \leq |\bar{x}|^p$, indicating that $\bar{t} \leq 0$, the choice is to set $x^* = 0$. Consequently, this results in $g(0, t) = (t - \bar{t})^2 = (t + |\bar{t}|)^2$. To meet the constraint $t^* \geq |x^*|^p$, the optimal selection is $t^* = 0$, which stands as the most suitable option for minimizing $g(0, t)$.

Algorithm 2 Non-convex projection ($p = \frac{s}{q} < 1$)

Algorithm 2 Non-convex projection $(p = \frac{s}{q} < 1)$

1: $\mathcal{R} \leftarrow \text{roots}\{a^{2q} + \frac{s}{q}(a^{2s} - \bar{t}a^s) - |\bar{x}|a^q\}$
2: $\bar{\mathcal{R}} \leftarrow \mathcal{R} \setminus \{\text{complex numbers and negative reals in } \mathcal{R}\}$
3: $\mathcal{T} \leftarrow \{(r^q, r^s) : r \in \bar{\mathcal{R}}\}$
4: $(\hat{x}, t^*) \leftarrow \text{argmin } \{g(x, t) : (x, t) \in \mathcal{T}\}$
5: $x^* \leftarrow \text{sign}(\bar{x})\hat{x}$

3.3 Convex projection

The convex projection for $\mathbf{y}$-update in (16) can be formulated as the following convex optimization problem:

$$\mathbf{y}^{k+1} = \underset{\mathbf{y}\in\mathcal{V}}{\operatorname{argmin}} \quad \left\| \mathbf{y} - \left(\mathbf{x}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho} \right) \right\|^2. \tag{27}$$

Convex problems can be solved by a variety of contemporary methods including bundle methods [34], sub-gradient projection [35], interior point methods [36], and ellipsoid methods [37]. The efficiency of optimization techniques relies mainly on exploiting the structure of the constraint set. As discussed in Sect. 1.3, our objective is to address the problem outlined in (5) with minimal assumptions on the set $\mathcal{V}$. Our only requirement is that $\mathcal{V}$ is a closed and convex set. Nevertheless, if feasible, one should capitalize on the inherent structure of $\mathcal{V}$ to potentially streamline the computational complexity involved in solving (27).

4 Rank minimization algorithm

We consider the problem in (1) and propose a method for approximating its solution efficiently. The Schatten-p heuristic of (1) can be written as:

$$\min_{\mathbf{X}\in\mathcal{M}} \quad \|\mathbf{X}\|_{p,*}^p \overset{\Delta}{=} \sum_{i=1}^{\mathrm{L}} |\sigma_i(\mathbf{X})|^p, \tag{28}$$

where $\mathrm{L} = \min\{\mathrm{m}, \mathrm{n}\}$ and $\sigma_i(\mathbf{X})$ is the ith singular value of $\mathbf{X}$. In the scenario where $p = 1$, (28) represents a convex problem, akin to the nuclear norm heuristic. We now consider a non-convex relaxation, specifically for the case where $0 < p < 1$. The problem in (28) attains an epi-graph form:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{t}} \quad & \mathbf{1}^\top \mathbf{t}, \\ \text{s.t.} \quad & |\sigma_i(\mathbf{X})|^p \le t_i, \quad i \in \{1, \dots \mathrm{L}\}, \quad \mathbf{X} \in \mathcal{M}, \end{aligned} \tag{29}$$

such that $\mathbf{t} = [t_i]_{i\in[\mathrm{L}]}$. Defining the epi-graph set $\mathcal{Y}$ for the function $\sigma(X)$, where $\mathcal{Y} \overset{\Delta}{=} \{(\sigma(\mathbf{X}), t) \in \mathbb{R}^2 : |\sigma(\mathbf{X})|^p \le t\} \subseteq \mathbb{R}^2$, the problem in (29) can be written as:

$$\min_{\mathbf{X},\mathbf{t}} \quad \mathbf{1}^\top \mathbf{t} + \mathbb{1}_{\mathcal{M}}(\mathbf{X}) + \sum_{i=1}^{\mathrm{L}} \mathbb{1}_{\mathcal{Y}}(\sigma_i(\mathbf{X}), t_i). \tag{30}$$

To formulate the problem in a manner amenable to ADMM, we introduce auxiliary variables, $\mathbf{Y} \in \mathbb{R}^{m\times n}$ and $\mathbf{z} = [z_i]_{i\in[\mathrm{L}]}$. This transformation leads to the following representation of the problem in (30):

$$\begin{aligned} \min_{\mathbf{X},\mathbf{t},\mathbf{Y},\mathbf{z}} \quad & \mathbf{1}^\top \mathbf{z} + \mathbb{1}_{\mathcal{M}}(\mathbf{Y}) + \sum_{i=1}^{\mathrm{L}} \mathbb{1}_{\mathcal{Y}}(\sigma_i(\mathbf{X}), t_i), \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{Y} : \boldsymbol{\Lambda}, \quad \mathbf{t} = \mathbf{z} : \boldsymbol{\theta}, \end{aligned} \tag{31}$$

where $\boldsymbol{\Lambda}$, $\boldsymbol{\theta}$ are the dual variables associated with $\mathbf{X}$ and $\mathbf{t}$ respectively. Similar to (9), the Lagrangian function associated with (31), augmented with a quadratic penalty for the equality constraint violation with a parameter $\rho > 0$, can be represented as:

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{X},\mathbf{Y},\mathbf{t},\mathbf{z},\boldsymbol{\Lambda},\boldsymbol{\theta}) &= \mathbf{1}^\top\mathbf{z} + \mathbb{1}_{\mathcal{M}}(\mathbf{Y}) + \sum_{i=1}^{\mathrm{L}} \mathbb{1}_{\mathcal{Y}}(\sigma_i(\mathbf{X}), t_i) \\ &+ Tr\{\boldsymbol{\Lambda}^\top(\mathbf{X}-\mathbf{Y})\} + \boldsymbol{\theta}^\top(\mathbf{t}-\mathbf{z}) + \frac{\rho}{2}(\|\mathbf{X}-\mathbf{Y}\|_{\mathrm{f}}^2 + \|\mathbf{t}-\mathbf{z}\|^2),\end{aligned} \tag{32}$$

where $Tr\{.\}$ is the trace operator. Given the 2-tuples $(\mathbf{X},\mathbf{t})$ and $(\mathbf{Y},\mathbf{z})$, the ADMM iterations are as follows:

$$(\mathbf{X},\mathbf{t})^{k+1} = \underset{\mathbf{X},\mathbf{t}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{X},\mathbf{Y}^k,\mathbf{t},\mathbf{z}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\theta}^k), \tag{33}$$

$$\mathbf{Y}^{k+1} = \underset{\mathbf{Y}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{X}^{k+1},\mathbf{Y},\mathbf{t}^{k+1},\mathbf{z}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\theta}^k), \tag{34}$$

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{X}^{k+1},\mathbf{Y}^{k+1},\mathbf{t}^{k+1},\mathbf{z},\boldsymbol{\Lambda}^k,\boldsymbol{\theta}^k), \tag{35}$$

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \rho(\mathbf{X}^{k+1} - \mathbf{Y}^{k+1}), \tag{36}$$

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \rho(\mathbf{t}^{k+1} - \mathbf{z}^{k+1}). \tag{37}$$

4.1 $(\mathbf{X},\mathbf{t})$ Update

By completing the square and employing some straightforward algebraic manipulations, it can be demonstrated that the problem described in (33) is equivalent to:

$$\begin{aligned}\min_{\mathbf{X},\mathbf{t}} \quad & \left\|\mathbf{X} - \bar{\mathbf{X}}^k\right\|_{\mathrm{f}}^2 + \left\|\mathbf{t} - \bar{\mathbf{t}}^k\right\|^2, \\ \text{s.t.} \quad & |\sigma_i(\mathbf{X})|^p \le t_i, \quad i \in \{1,\dots \mathrm{L}\},\end{aligned} \tag{38}$$

where $\bar{\mathbf{X}}^k \triangleq \mathbf{Y}^k - \frac{\boldsymbol{\Lambda}^k}{\rho}$ and $\bar{\mathbf{t}}^k \triangleq \mathbf{z}^k - \frac{\boldsymbol{\theta}^k}{\rho}$. For simplicity, we will omit the iteration index k. Let's assume that $\mathbf{X} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}^\top$ and $\bar{\mathbf{X}} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top$ represent the Singular Value Decomposition (SVD) of $\mathbf{X}$ and $\bar{\mathbf{X}}$, respectively. Here, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Delta}$ are diagonal matrices with the singular values associated with $\mathbf{X}$ and $\bar{\mathbf{X}}$, while $\mathbf{P}$, $\mathbf{U}$, $\mathbf{Q}$, and $\mathbf{V}$ are unitary matrices. Following the steps in [38, Theorem 3], we can express the first term of (38) as:

$$\begin{aligned}\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|_{\mathrm{f}}^2 &= \left\|\mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}^\top - \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top\right\|_{\mathrm{f}}^2 \\ &= \left\|\mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}^\top\right\|_{\mathrm{f}}^2 + \left\|\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top\right\|_{\mathrm{f}}^2 - 2T\{\mathbf{X}^\top\bar{\mathbf{X}}\} \\ &\overset{(a)}{=} Tr\{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\} + Tr\{\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\} - 2Tr\{\mathbf{Q}\boldsymbol{\Sigma}^\top\mathbf{P}^\top\mathbf{U}\boldsymbol{\Delta}\mathbf{V}^\top\} \\ &\overset{(b)}{\ge} Tr\{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\} + Tr\{\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\} - 2Tr\{\boldsymbol{\Sigma}^\top\boldsymbol{\Delta}\} = \|\boldsymbol{\Sigma} - \boldsymbol{\Delta}\|_{\mathrm{f}}^2,\end{aligned} \tag{39}$$

where (a) is because $\mathbf{P}^\top\mathbf{P} = \mathbf{Q}^\top\mathbf{Q} = \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_{\mathrm{L}\times\mathrm{L}}$ with $\mathbf{I}_{\mathrm{L}\times\mathrm{L}}$ being an identity matrix of size L, and exploiting the circular property of the trace while (b) holds is from the main result of [39]. In order to make $\left\|\mathbf{X} - \bar{\mathbf{X}}^k\right\|_{\mathrm{f}}^2$ achieve its derived lower bound, we set $\mathbf{P} = \mathbf{U}$ and $\mathbf{Q} = \mathbf{V}$.

The problem in (38) is then equivalent to:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{t}} \quad & \|\mathbf{x} - \bar{\mathbf{x}}\|^2 + \left\|\mathbf{t} - \bar{\mathbf{t}}\right\|^2, \\ \text{s.t.} \quad & |x_i|^p \le t_i, \quad i \in \{1, \dots \mathrm{L}\}, \end{aligned} \tag{40}$$

where $\mathbf{x} = [x_i]_{i\in[\mathrm{L}]}$ and $\bar{\mathbf{x}} = [\bar{x}_i]_{i\in[\mathrm{L}]}$ are the vectors of singular values of the matrices $\mathbf{X}$ and $\bar{\mathbf{X}}$ respectively. The optimal solution $\mathbf{X}^*$ for (38) can be determined by first finding the optimal $\mathbf{x}^*$ for (40), and then obtaining $\mathbf{X}^* = \mathbf{U}\boldsymbol{\Sigma}^*\mathbf{V}^\top$, where $\boldsymbol{\Sigma}^* = \mathrm{diag}(\mathbf{x}^*)$ and diag(.) denotes an operator that transforms a vector into its corresponding diagonal matrix. Given that the problem in (40) is separable, we will proceed by omitting the index i and focus solely on solving:

$$\min_{x,t} \quad (x - \bar{x})^2 + (t - \bar{t})^2, \quad \text{s.t.} \quad |x|^p \le t. \tag{41}$$

It can be realized that (41) is similar to (18), hence, its optimal solution can be found by applying Algorithm 2.

4.2 $(\mathbf{Y}, \mathbf{z})$ update

Upon updating $(\mathbf{X}, \mathbf{t})$ with $\boldsymbol{\Lambda}$ and $\boldsymbol{\theta}$ held constant, the problem in (34) can be reformulated as:

$$\mathbf{Y}^{k+1} = \underset{\mathbf{Y}\in\mathcal{M}}{\mathrm{argmin}} \left\|\mathbf{Y} - \left(\mathbf{X}^{k+1} + \frac{\boldsymbol{\Lambda}^k}{\rho}\right)\right\|_{\mathrm{f}}^2, \tag{42}$$

which is clearly a convex optimization problem representing the projection of the point $\mathbf{X}^{k+1} + \frac{\boldsymbol{\Lambda}^k}{\rho}$ on the set $\mathcal{M}$ and can be solved by various known class of algorithms as discussed in Sect. 3.3. Following the update of $\mathbf{Y}$, the update for $\mathbf{z}$ in (35) is as follows:

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\mathrm{argmin}} \quad \mathbf{1}^\top\mathbf{z} + \frac{\rho}{2}\left\|\mathbf{z} - \left(t^{k+1} + \frac{\boldsymbol{\theta}^k}{\rho}\right)\right\|^2, \tag{43}$$

which results in a closed-form solution for $\mathbf{z}^{k+1} = \mathbf{t}^{k+1} + \frac{\boldsymbol{\theta}^k - \mathbf{1}}{\rho}$.

5 Proximal gradient algorithm

The pQN-ADMM algorithm adeptly handles the ℓ_p relaxation of (2), refraining from assuming any specific structure for $\mathcal{V}$ beyond its closed and convex nature. Primarily, the algorithm hinges on the computation of Euclidean projections onto $\mathcal{V}$, as outlined in (27).

In this section, we consider a sub-class of problems with a specific structure for the convex set of the form $\mathcal{V} = \{\mathbf{x} : f(\mathbf{x}) \le 0\}$, where $f(\mathbf{x})$ is a convex function with Lipschitz continuous gradient. i.e., f is L-smooth: $\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\right\| \le L\left\|\mathbf{x} - \mathbf{y}\right\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Specifically, in order to solve:

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_p^p, \quad \text{s.t.} \quad f(\mathbf{x}) \leq 0, \tag{44}$$

we aim to develop an efficient algorithm with some convergence guarantees for the following Lagrangian relaxation:

$$\min_{\mathbf{x}} \quad F(\mathbf{x}) \triangleq \|\mathbf{x}\|_p^p + \frac{\mu}{2} f(\mathbf{x}), \tag{45}$$

where $\mu \geq 0$ is the dual multiplier that captures the trade-off between solution sparsity and fidelity. It is imperative to acknowledge that (44) and (45) exhibit a relationship, albeit not being strictly equivalent.

A canonical problem for the regularized risk minimization has the following form:

$$\min_{\mathbf{x}} \quad g(\mathbf{x}) + h(\mathbf{x}), \tag{46}$$

where h is an L-smooth loss function, and g represents the regularizer term. In cases where both g and h exhibit convexity, the Proximal Gradient (PG) algorithm [40] can iteratively compute a solution to (46) through PG steps.

$$\mathbf{x}^{k+1} = \mathbf{prox}_{g/\lambda}(\mathbf{x}^k - \nabla h(\mathbf{x}^k)/L), \tag{47}$$

where $\mathbf{prox}_{g/\lambda}(.) \triangleq \underset{\mathbf{x}}{\operatorname{argmin}}\, g(\mathbf{x}) + \frac{\lambda}{2}\|\mathbf{x} - \cdot\|^2$, for some constant λ. When g is convex, the proximal map $\mathbf{prox}_{g/\lambda}$ is well-defined, thus, the PG step can be computed.

In comparing both (45) and (46), it is observed that the convexity assumption of $g(\mathbf{x})$ in (46) is not met for $\|\mathbf{x}\|_p^p$ in (45). When the regularizer is a continuous non-convex function, the proximal map $\mathbf{prox}_{g/\lambda}$ may not exist, and computing it in closed form becomes a challenging task.

On the contrary, in the case of $\|\mathbf{x}\|_p^p$, leveraging similar reasoning as employed in the non-convex projection step introduced in Sect. 3.2, our objective is to derive an analytical solution that can be efficiently computed. Specifically, assuming $p \in (0, 1)$ is a positive rational number, the proposed method for computing the proximal map of $\|\mathbf{x}\|_p^p$ involves finding the roots of a polynomial of order $2q$, where $q \in \mathbb{Z}_+$ such that $p = s/q$ for some $s \in \mathbb{Z}_+$.

Since f is L-smooth, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we have:

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{L}{2}\left\|\mathbf{x} - \mathbf{y}\right\|^2. \tag{48}$$

Given $\mathbf{x}^k$, replacing $f(\mathbf{x})$ with the upper bound in (48) for $\mathbf{y} = \mathbf{x}^k$, the prox-gradient operation naturally arises as follows:

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}}\, \|\mathbf{x}\|_p^p + \frac{\mu}{2}\left[f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^\top (\mathbf{x} - \mathbf{x}^k) + \frac{L}{2}\left\|\mathbf{x} - \mathbf{x}^k\right\|^2\right]. \tag{49}$$

By completing the square, (49) yields to:

$$\mathrm{x}^{k+1} = \arg\min_{\mathrm{x}} \|\mathrm{x}\|_p^p + \frac{\mu L}{4}\left\|\mathrm{x} - \left(\mathrm{x}^k - \frac{1}{L}\nabla f(\mathrm{x}^k)\right)\right\|^2. \tag{50}$$

Defining $\bar{\mathrm{x}}^k \triangleq \mathrm{x}^k - \frac{1}{L}\nabla f(\mathrm{x}^k)$, (50) can be rewritten as:

$$\begin{aligned}\mathbf{x}^{k+1} &= \operatorname*{argmin}_{\mathbf{x}} \|\mathbf{x}\|_p^p + \frac{\mu L}{4}\left\|\mathbf{x} - \bar{\mathbf{x}}^k\right\|^2 \\ &= \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{n} |x_i|^p + \frac{\mu L}{4}(x_i - \bar{x}_i^k)^2,\end{aligned} \tag{51}$$

which is clearly a separable structure in the entries of $\mathbf{x}$. Therefore, for each $i \in [n]$, we have:

$$x_i^{k+1} = \operatorname*{argmin}_{x_i} |x_i|^p + \frac{\mu L}{4}(x_i - \bar{x}_i^k)^2 = \mathbf{prox}_{\bar{g}/\frac{\mu L}{2}}(\bar{x}_i^k), \tag{52}$$

where $\bar{g} : \mathbb{R} \to \mathbb{R}_+$ such that $\bar{g}(t) = |t|^p$ for some positive rational $p \in (0, 1)$.

Next, we consider a generic form of (52), i.e., given some $\bar{t} \in \mathbb{R}$, we would like to compute:

$$t^* = \operatorname*{argmin}_{t} \left\{ |t|^p + \frac{\mu L}{4}(t - \bar{t})^2 \right\}. \tag{53}$$

The first-order optimality condition for (53) can be written as:

$$p|t^*|^{p-1}\mathrm{sign}(t^*) + \frac{\mu L}{2}(t^* - \bar{t}) = 0. \tag{54}$$

Using similar arguments as in Sect. 3.2, we can conclude that the optimal solution t^* attains the property that $\mathrm{sign}(t^*) = \mathrm{sign}(\bar{t})$. Without loss of generality, exploiting the symmetry of the function $\bar{g}$, we only consider the case when $\bar{t} > 0$; hence, the optimal solution t^* is the smallest positive root of the following polynomial:

$$p|t^*|^{p-1} + \frac{\mu L}{2}(t^* - \bar{t}) = 0. \tag{55}$$

Similar to (26), suppose $0 < p = \frac{s}{q} < 1$ for some positive integers s and q. By employing the variable transformation $a \triangleq (t^*)^{\frac{1}{q}}$, the optimality condition in (55) is simplified to the task of finding the roots of a polynomial of degree $2q$:

$$a^{2q} - \bar{t}a^q + \frac{2s}{q\mu L}a^s = 0. \tag{56}$$

Algorithm 3 Accelerated PG algorithm

Algorithm 3 Accelerated PG algorithm

1: Initialize: μ, $s = 1$, $q = 2$, l, $\mathbf{x}^0$, $\mathbf{x}^1$, $k = 1$.
2: **repeat**
3: $\quad \mathbf{y}^k = \mathbf{x}^k + \frac{k-1}{k+2}(\mathbf{x}^k - \mathbf{x}^{k-1})$
4: $\quad \Delta^k = \max_{t=\max\{1,k-l\},\ldots,k} F(\mathbf{x}^t)$
5: $\quad$ **if** $F(\mathbf{y}^k) \leq \Delta^k$ **then**:
6: $\quad\quad \mathbf{v}^k = \mathbf{y}^k$
7: $\quad$ **else**:
8: $\quad\quad \mathbf{v}^k = \mathbf{x}^k$
9: $\quad \bar{\mathbf{x}}^k = \mathbf{v}^k - \frac{1}{L}\nabla f(\mathbf{v}^k)$
10: $\quad$ **for** $i \in [n]$ **do**:
11: $\quad\quad$ `solve` $a^{2q} - \bar{x}_i a^q + \frac{2s}{q\mu L} a^s = 0$
12: $\quad\quad x_i^{k+1} = a^{*q}$
13: $\quad k = k + 1$
14: **until** convergence

In order to solve (44) effectively, we will employ Algorithm 3, which implements the non-convex inexact Accelerated Proximal Gradient (APG) descent method as presented in [41, Algorithm 2]. In summary, Algorithm 3 is designed to tackle composite problems of the form in (46), making the assumptions that h is L-smooth and g is a proper lower-semicontinuous function such that $F \triangleq h + g$ is bounded from below and coercive. This means that $\lim_{\|\mathbf{x}\|\to\infty} F(\mathbf{x}) = +\infty$. It is important to note that neither h nor g are required to be convex. Algorithm 3 can be summarized as follows:

- An extrapolation $\mathbf{y}_k$ is generated as introduced in [42] for the APG algorithm (step 3).
- Steps 4 through 9 encompass a mechanism for a non-monotone update of the objective function. Specifically, $F(\mathbf{y}_k)$ undergoes scrutiny concerning its relation to the maximum among the most recent l objective values. Step 9 is responsible for adjusting the gradient step accordingly. This adjustment occasionally allows $\mathbf{y}^k$ to increase the objective, resulting in a situation where $F(\mathbf{y}^k)$ becomes lower than the maximum objective value observed in the latest l iterations.
- Steps 11 and 12 represent the solution of the PG step using the non-convex projection method.

In the next part, we show that Algorithm 3 converges to a critical point and it exhibits a convergence rate of $O(\frac{1}{K})$, where K is the iteration budget that is given to the algorithm.

Definition 1 ([43]) The Frechet sub-differential of F at $\mathbf{x}$ is

$$\hat{\partial}F(\mathbf{x}) \triangleq \Big\{ \mathbf{u} : \lim_{\mathbf{y}\neq\mathbf{x}} \lim_{\mathbf{y}\to\mathbf{x}} \frac{F(\mathbf{y}) - F(\mathbf{x}) - \mathbf{u}^{\top}(\mathbf{y}-\mathbf{x})}{\|\mathbf{y}-\mathbf{x}\|} \geq 0 \Big\}. \tag{57}$$

The sub-differential of F at $\mathbf{x}$ is

$$\partial F(\mathbf{x}) \triangleq \{ \mathbf{u} : \exists \mathbf{x}^k \to \mathbf{x}, F(\mathbf{x}^k) \to F(\mathbf{x}) \text{ and } \mathbf{u}^k \in \hat{\partial}F(\mathbf{x}^k) \to \mathbf{u} \text{ as } k \to \infty \}. \tag{58}$$

Definition 2 [43] $\mathbf{x}$ is a critical point of F if $0 \in \partial g(\mathbf{x}) + \nabla h(\mathbf{x})$.

By comparing (46) and (45), it can be realized that the functions $g(\mathbf{x})$ and $h(\mathbf{x})$ in definition 2 are equal to $\|\mathbf{x}\|_p^p$ and $\frac{\mu}{2} f(\mathbf{x})$, respectively.

Theorem 2 *The sequence $\mathbf{x}^k$ generated from Algorithm 3 has at least one limit point and all the generated limit points are critical points of (45). Moreover, the algorithm converges with rate $O(\frac{1}{K})$, where K is the iteration budget given to the algorithm.*

Proof It can easily be verified that our problem in (45) satisfies all required assumptions for Algorithm 3. Indeed,

1. The function $g(\mathbf{x}) = \|\mathbf{x}\|_p^p$ is a proper and lower semi-continuous function.
2. The gradient of $h(\mathbf{x}) = \frac{\mu}{2} f(\mathbf{x})$ is $\bar{L}$-Lipschitz smooth, i.e., $\|\nabla h(\mathbf{x}) - \nabla h(\mathbf{y})\| \leq \bar{L}\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, with $\bar{L} = \frac{\mu}{2} L$.
3. $F(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$ is bounded from below, i.e., $F(\mathbf{x}) \geq 0$.
4. $\lim_{\|\mathbf{x}\|\to\infty} F(\mathbf{x}) = \infty$.
5. Let $\mathcal{G}(\mathbf{x}) \triangleq \mathbf{x} - \mathbf{prox}_{g/\lambda}(\mathbf{x} - \nabla h(\mathbf{x})/L)$. From [43, 44], $\|\mathcal{G}(\mathbf{x})\|^2$ can be used to measure how far $\mathbf{x}$ is from optimality. Specifically, $\mathbf{x}$ is a critical point of (46) if and only if $\mathcal{G}(\mathbf{x}) = 0$.
6. The introduced non-convex projection method is an exact solution for the proximal gradient step. This is because it is based on finding the roots of a polynomial of order $2q$ in (56).

Therefore, from Theorem 4.1 and Proposition 4.3 of [41], the sequence generated by Algorithm 3 converges to a critical point of (45). Additionally, $\|\mathcal{G}(\mathbf{x}^k)\|^2$ converges with rate $O(\frac{1}{K})$, thereby completing the proof. □

Remark 1

The global convergence of several exact iterative methods that solve (46) has been explored, under the framework of Kurdyka-Lojasiewicz (KL) theory, in various additional literature including [43, 45–48]. Other work (see [49] and references therein) considered the linear convergence of non-exact algorithms with relaxations on the assumptions of KL theory, however, it is difficult to verify that the sequence generated by Algorithm 3 satisfies the relaxed assumptions stated in [49].

6 Numerical results

In this section, we present numerical examples to illustrate the application of the pQN-ADMM algorithm, as expounded in Algorithm 1, and the non-convex projection method delineated in Algorithm 2. Within each of the ensuing examples, we conduct comparative analyses with the convex ℓ_1 relaxation solution, achieved through the use of the MOSEK solver [50], and alternative $\ell_{0.5}$-based solutions previously proposed in the literature.

The degree of the polynomial for which the roots are determined during the non-convex projection step depends on the value of q in the context of $p = \frac{s}{q}$. It might lead one to speculate that the computational complexity of the non-convex projection step is contingent on the specific value of p, suggesting that lower values of p result in slower algorithm performance. In order to explore this aspect, we systematically performed the non-convex projection step 200 times on a vector of 1024 elements, as part of a sparse vector reconstruction example. Throughout this process, we systematically varied the values of the parameter p, considering a range of p values, specifically $p \in \{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\}$. The average time to perform the non-convex projection for the entire vector, where the roots of (26) for each p are computed using the "root" command in MATLAB, is observed to be nearly constant, approximately 0.03 s. Furthermore, our numerical experiments in this particular example indicated that for $p \in \{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}\}$, no substantial improvement over the $\ell_{0.5}$ case was observed. As a result, these cases are currently undergoing further investigation and are not included in the numerical results section.

6.1 Sparse vector recovery (SVR)

In this section, we implement a sparse vector reconstruction problem and compare the solution of the pQN-ADMM algorithm with the ℓ_1 convex relaxation along with an $\ell_{0.5}$ relaxation solution and Linear Approximation for Index Tracking (LAIT), as presented in [51] and [52], respectively.

Let $n = 2^{10}$ and $m = n/4$, randomly construct the sparse binary matrix, $\mathbf{M} \in \mathbb{R}^{m \times \frac{n}{2}}$, with a few number of ones in each column. The number of ones in each column of $\mathbf{M}$ is generated independently and randomly in the range of integers between 10 and 20, and their locations are randomly chosen independently for each column. Let $\mathbf{U} = [\mathbf{M}, -\mathbf{M}]$, which is the vertical concatenation of the matrix $\mathbf{M}$ and its negative. Following the same setup in [53], the column orthogonality in $\mathbf{U}$ is not satisfied. Let $\mathbf{x}_{\text{opt}} \in \mathbb{R}^n$ be a reference signal with $\|\mathbf{x}_{\text{opt}}\|_0 = \lceil 0.2n \rceil$, where the non-zero locations are chosen uniformly at random with the values following a zero mean, unit variance Gaussian distribution. Let $\mathbf{v} = \mathbf{U}\mathbf{x}_{\text{opt}} + \mathbf{n}$ be the allowable measurement, where $\mathbf{n} \in \mathbb{R}^m$ is a Gaussian random vector with zero mean and co-variance matrix $\sigma^2 \mathbf{I}_{m \times m}$, where $\mathbf{I}$ is the identity matrix. The sparse vector is reconstructed from $\mathbf{v}$ by solving (5) with $\mathcal{V} = \{\mathbf{x} : \|\mathbf{U}\mathbf{x} - \mathbf{v}\|/\|\mathbf{v}\| - \epsilon \leq 0\}$, where $\epsilon = \frac{3\sigma}{\|\mathbf{v}\|}$. All the algorithms are terminated if $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|/\|\mathbf{x}^{k-1}\| \leq 10^{-4}$ or a budget of 200 iterations is consumed.

Figure 1 depicts the correlation between sparsity levels and noise variances concerning solutions derived through ℓ_1 norm minimization, $\ell_{0.5}$, LAIT, and pQN-ADMM techniques. A threshold of 10^{-6} was imposed, designating entries of the solution vector as zero if they fell below this threshold. The reported outcomes are based on the average results obtained from 20 independently conducted random

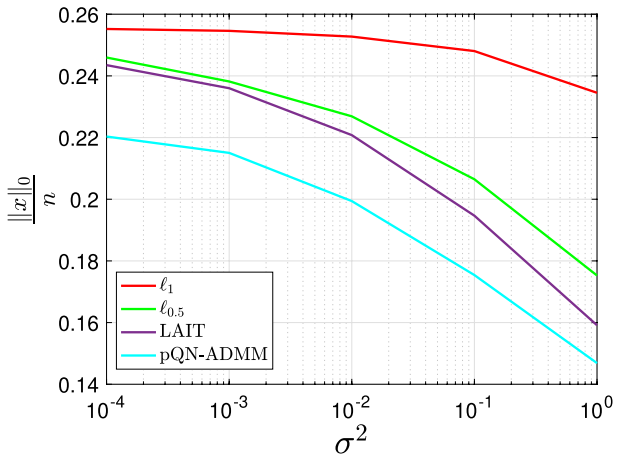


Fig. 1 The influence of noise variance on the sparsity of solutions generated by ℓ_1 norm, $\ell_{0.5}$, LAIT, and the pQN-ADMM

iterations. Notably, it becomes evident that the pQN-ADMM algorithm consistently yields solutions with higher sparsity levels in comparison to its counterpart baseline methods, across a range of σ^2 values. As σ^2 increases, the sparsity level for all approaches decreases, attributable to the heightened scarcity of information pertaining to the original signal within the realization vector, thereby compromising the precision of the reconstruction process.

6.2 Rank minimization problem (RMP)

Within this section, our primary focus is directed towards the exploration of the pQN-ADMM algorithm within the RMP framework, as presented in Sect. 4. We commence by engaging in a matrix completion scenario, presenting an extensive comparative analysis pitting the pQN-ADMM algorithm against various baseline methods rooted in the Schatten-p quasi-norm framework.

Additionally, we delve into a time domain system identification example. Notably, we restrict our comparative analysis to the convex nuclear norm. This singularity in focus arises from the unique constraint nature of the problem at hand, specifically the Hankel constraint. To the best of our knowledge, there are no other Schatten-p-based algorithms capable of addressing constraints of this specific nature in the proposed formulation. This serves to underscore the remarkable versatility of the pQN-ADMM algorithm in handling a broad spectrum of constraints, be they within the vector or matrix domain.

6.2.1 Matrix completion

In this section, we apply our algorithm (pQN-ADMM) to a matrix completion example and compare the result to the Matrix Iterative Re-weighted Least Squares (Matrix-IRLS) [54, 55], truncated Iterative Re-weighted unconstrained Lq (tIRucLq) [56] and Iterative Re-weighted Least Squares (sIRLS-p & IRLS-p) [57] algorithms. The matrix completion problem is a special case of the low-rank minimization where a linear transform takes a few random entries of an ambiguous matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. Given only

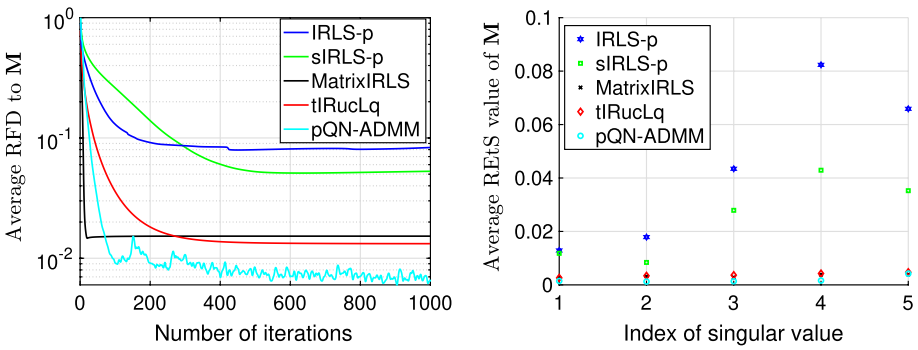
these entries, the goal is to approximate $\mathbf{X}$ and find the missing ones. The matrix completion problem with low-rank recovery can be approximated by,

$$\min_{\mathbf{X}} \quad \|\mathbf{X}\|_{p,*}^{p}, \quad \text{s.t.} \quad \|\mathcal{A}(\mathbf{X}) - \mathbf{b}\| \leq \epsilon, \tag{59}$$

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^{q}$ is a linear map with $q \ll mn$ and $\mathbf{b} \in \mathbb{R}^{q}$. To facilitate the application of the aforementioned algorithms, the linear transform $\mathcal{A}(\mathbf{X})$ will be reformulated as $\mathbf{A}\text{vec}(\mathbf{X})$, where $\mathbf{A} \in \mathbb{R}^{q\times mn}$ and $\text{vec}(\mathbf{X}) \in \mathbb{R}^{\text{mn}}$ represents a vector obtained by stacking the columns of the matrix $\mathbf{X}$.

A random matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ with rank r is created using the following method: 1) $\mathbf{M} = \mathbf{M}_{\text{L}}\mathbf{M}_{\text{R}}^{\top}$, where $\mathbf{M}_{\text{L}} \in \mathbb{R}^{m\times \text{r}}$ and $\mathbf{M}_{\text{R}} \in \mathbb{R}^{n\times \text{r}}$. 2) The entries of both $\mathbf{M}_{\text{L}}$ and $\mathbf{M}_{\text{R}}$ are i.i.d Gaussian random variables with zero mean and unit variance. Let $\hat{\mathbf{M}} = \mathbf{M} + \mathbf{Z}$, where $\mathbf{Z} \in \mathbb{R}^{m\times n}$ is a Gaussian noise with each entry being an i.i.d Gaussian random variable with zero mean and variance σ^2. The vector $\mathbf{b}$ is then created by selecting random q elements from $\text{vec}(\hat{\mathbf{M}})$. Since $\mathbf{b} = \mathbf{A}\text{vec}(\hat{\mathbf{M}})$, one can easily construct the matrix $\mathbf{A}$ which is a sparse matrix where each row is composed of a value 1 at the index of the corresponding selected entry in the vector $\mathbf{b}$ while the rest are zeros. We set $m = n = 100$, $\text{r} = 5$ and $p = 0.5$. Let $\text{d}_{\text{r}} = \text{r}(\text{m} + \text{n} - \text{r})$ denotes the dimension of the set of rank r matrices and define $\text{s} = \frac{\text{q}}{\text{mn}}$ as the sampling ratio. We assume that $\text{s} = 0.195$ which yields to $q = 1950$. It can be realized that $\frac{\text{d}_{\text{r}}}{q} < 1$. We set $\sigma = 0.1$, $\epsilon = 10^{-3}$, and let the algorithms terminate if a budget of 1000 iterations is reached. To compare the solutions across different algorithms, where $\mathbf{X}^*$ represents the solution for (59), we evaluate the average of 50 runs based on two metrics: a) the Relative Frobenius Distance (RFD) to the matrix $\mathbf{M}$, defined as $RFD = \frac{\|\mathbf{X}^*-\mathbf{M}\|_{\text{f}}}{\|\mathbf{M}\|_{\text{f}}}$, and b) the Relative Error to Singular (REtS) values of $\mathbf{M}$, given by $RES_i = \frac{|\sigma_i(\mathbf{X}^*)-\sigma_i(\mathbf{M})|}{\sigma_i(\mathbf{M})}$ for $i \in [\min\{m, n\}]$.

In Fig. 2a, b, we report the average RFD and REtS values for all the algorithms. Despite that, all the baselines are designed to exploit the specific structure of the matrix completion problem, described in (59), while the proposed pQN-ADMM doesn't, it is competitive against them all. This in turn shows the effectiveness of the pQN-ADMM algorithm in solving the rank minimization problems without requiring any prior information about the structure of the associated convex set.



(a) RFD to $\mathbf{M}$. (b) REtS values of $\mathbf{M}$.

Fig. 2 The RFD and REtS average values

6.2.2 Time domain system identification

We consider a stable Single Input Single Output (SISO) system operating in discrete time, wherein the input vector $\mathbf{u} \in \mathbb{R}^T$ corresponds to a temporal span denoted by T, representing the number of input samples. The system is characterized by an impulse response consisting of a fixed number of samples denoted as n. The resultant output of the system is represented by $\mathbf{y} \in \mathbb{R}^m$. However, in practical scenarios, only noisy realizations, denoted as $\hat{\mathbf{y}}$, are observable. This realization is expressed as $\hat{\mathbf{y}} \overset{\Delta}{=} \mathbf{y} + \mathbf{z} = \mathbf{h} \circledast \mathbf{u} + \mathbf{z}$, where $\mathbf{h} \in \mathbb{R}^n$ signifies the system's original impulse response, $\mathbf{z} \in \mathbb{R}^m$ is a random vector with entries drawn independently from a uniform distribution within the range $[-0.25, 0.25]$, and $\circledast$ denotes the convolution operator.

Exploiting the window property of convolution, which asserts that $m = n + T - 1$, we establish the relationship among the components u_i, h_i, and y_i through the linear convolution relation $y_i = \sum_{j=-\infty}^{\infty} h_j u_{i-j}$. Herein, u_i, h_i, and y_i represent the ith components of the vectors $\mathbf{u}$, $\mathbf{h}$, and $\mathbf{y}$, respectively. To succinctly represent the convolution, let $\mathbf{T} \in \mathbb{R}^{m \times n}$ be the Toeplitz matrix formed by the input $\mathbf{u}$, allowing us to express $\mathbf{h} \circledast \mathbf{u} = \mathbf{h}\mathbf{T}^\top$. Furthermore, assuming $\mathbf{x} \in \mathbb{R}^n$ to be an impulse response variable, we introduce $\mathbf{X} \in \mathbb{R}^{n \times n}$ as a Hankel matrix formed by the entries of $\mathbf{x}$. From [58–61], the minimum order time domain system identification problem can be formulated as:

$$\min_{\mathbf{X}} \quad \mathbf{Rank}(\mathbf{X}), \tag{60a}$$

$$\text{s.t.} \quad \mathbf{X} = \mathit{Hankel}(\mathbf{x}), \tag{60b}$$

$$\left\| \hat{\mathbf{y}} - \mathbf{x}\mathbf{T}^\top \right\|^2 \leq \epsilon, \tag{60c}$$

(60b) ensures that $\mathbf{X}$ is a Hankel matrix and (60c) holds to make the result by applying the input, $\mathbf{u}$, to the optimal impulse response, $\mathbf{x}$, fit the available noisy data, $\hat{\mathbf{y}}$, in a non-trivial sense. Defining the convex set $\mathcal{C} \overset{\Delta}{=} \{\mathbf{X} \in \mathbb{R}^{n \times n} : \left\| \hat{\mathbf{y}} - \mathbf{x}\mathbf{T}^\top \right\|^2 - \epsilon \leq 0, \mathbf{X} = \mathit{Hankel}(\mathbf{x})\}$, (60) can be cast as:

$$\min_{\mathbf{X} \in \mathcal{C}} \quad \mathbf{Rank}(\mathbf{X}), \tag{61}$$

which is clearly identical to the problem in (1). The problem was solved using the same pQN-ADMM approach discussed in Sect. 4.

Let $T = m = 50$ and $n = 40$. It is pertinent to note that $m < T + n - 1$, is a reasonable assumption aligning with practical applications where only a specific window is available to observe the output. The simulation is conducted across 5 distinct original system orders denoted by $\eta \in \{2, 4, 6, 8, 10\}$. An input vector, $\mathbf{u}$, is generated, with its elements being independent and following a uniform distribution over the interval $[-5, 5]$. For each η:

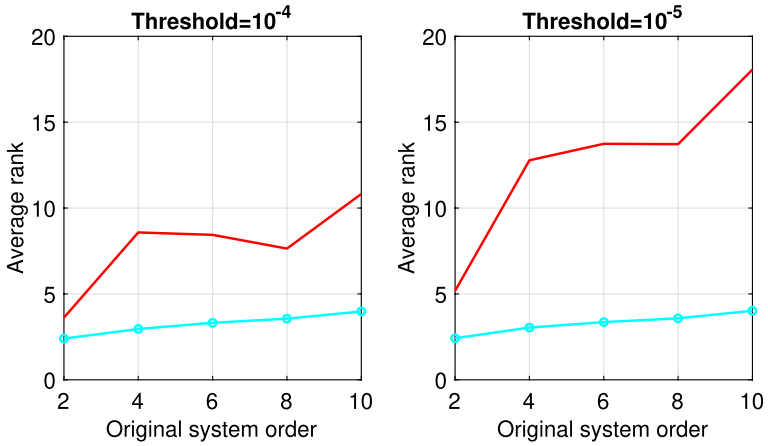


Fig. 3 Average rank vs. original system order. Red and cyan colors are for the nuclear norm and pQN-ADMM algorithm, respectively

1 Fifty random stable systems are generated using the 'drss' command in MATLAB.
2 The generated input is applied to each system, yielding the corresponding noisy output $\hat{\mathbf{y}}$.
3 Given the output $\hat{\mathbf{y}}$, the problem specified in (60) is solved, and the rank of the corresponding system is computed using singular value decomposition.
4 The obtained results are averaged to derive the corresponding average rank for each original η.

Figure 3 presents the average rank results obtained through the nuclear norm and pQN-ADMM heuristics. The outcomes correspond to two distinct threshold values, wherein the threshold is defined as the value below which the singular value is considered zero. Notably, the introduced pQN-ADMM approach demonstrates superior performance compared to the nuclear norm heuristic for both threshold values. Furthermore, as the threshold value decreases from 10^{-4} to 10^{-5}, the pQN-ADMM's behavior remains consistent, while the average rank for the nuclear norm exhibits an increase. This observation underscores the robustness of the derived pQN-ADMM relative to the nuclear norm approach.

Table 1 provides the standard deviation values for the algorithms. It is evident that the standard deviation remains constant for the pQN-ADMM when altering the threshold; conversely, it increases for the nuclear norm as the threshold value decreases.

Table 1 Standard deviation for different threshold values

	Threshold $= 10^{-4}$			**Threshold** $= 10^{-5}$		
	η**=2**	η**=6**	η**=10**	η**=2**	η**=6**	η**=10**
Nuclear norm	2.3907	6.6668	7.2572	6.9877	11.2638	11.7854
pQN-ADMM	0.5292	0.9042	1.0861	0.5325	0.9113	1.0861

6.3 Accelerated proximal gradient (APG) algorithm

In this section, we present numerical results for the APG method, as outlined in Algorithm 3. Our primary objective is to address the minimization problem (45) with $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$.

Consistent with the approach in [62], we initiate the process by generating the target signal $\mathbf{x}^*$ through:

$$x_i^* = \begin{cases} \Theta_i^{(1)} 10^{3\Theta_i^{(2)}}, & \forall\, i \in \Lambda, \\ 0, & \forall\, i \in [n] \setminus \Lambda; \end{cases} \tag{62}$$

where the design parameters $\Lambda \subset [n]$, and $\Theta_i^{(1)}, \Theta_i^{(2)}$ for $i \in \Lambda$ are chosen as follows:

1 The index set $\Lambda \subset [n]$ is constructed by selecting a subset of $[n]$ with cardinality s uniformly at random;
2 $\{\Theta_i^{(1)}\}_{i\in\Lambda}$ are Independent and Identically Distributed (IID) Bernoulli random variables taking values ± 1 with equal probability;
3 $\{\Theta_i^{(2)}\}_{i\in\Lambda}$ are IID uniform $[0, 1]$ random variables.

The measurement matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is constructed as a partial Discrete Cosine Transform (DCT) matrix, with its rows corresponding to $m < n$ frequencies. Specifically, these m indices are selected uniformly at random from the set $[n]$. The noisy measurement vector $\mathbf{b} \in \mathbb{R}^m$ is subsequently defined as $\mathbf{b} = \mathbf{A}(\mathbf{x}^* + \boldsymbol{\epsilon}_1) + \boldsymbol{\epsilon}_2$, where $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are IID random vectors with entries following zero mean Gaussian distributions with variances σ_1^2 and σ_2^2, respectively.

In our experiments, $n = 4096$, $s = \lceil 0.5m \rceil$ and the APG algorithm memory to 5, i.e., $l = 5$ in Algorithm 3. Following the medium noise setup in [63], we set $\sigma_1 = 0.005$, $\sigma_2 = 0.001$.

For the objective function $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$, the Lipschitz constant is given by $L = 2\|\mathbf{A}\|^2$. Our experimental design encompasses varying values of m, representing the number of noisy measurements, and μ, serving as the trade-off parameter in (45). For each unique combination of (m, μ), we conduct 20 random instances of the triplet $(\mathbf{x}^*, \mathbf{A}, \mathbf{b})$ to account for the inherent statistical variability of the problem. Each random instance is subsequently solved using Algorithm 3, and the average performance is reported. The termination criterion for Algorithm 3 is defined as the relative error between consecutive iterates satisfies $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| / \|\mathbf{x}^{k-1}\| \le 10^{-5}$.

In our experiments, we conducted a comparative analysis of solving (45) for $p = 0.5$ against $p = 1$, corresponding to ℓ_1-optimization for sparse recovery. Specifically, for $p = 0.5$, denoting $\ell_{0.5}$ minimization, we employed Algorithm 3, referred to as $\ell_{0.5}$ exact. Additionally, we utilized Algorithm 2 from [64], denoted as $\ell_{0.5}$ approx. Conversely, for $p = 1$, where the ℓ_1-minimization problem is convex, we employed the FISTA algorithm from [42]. The solutions are denoted as $\bar{\mathbf{x}}$, while the target signal, derived from (62), is denoted as $\mathbf{x}^*$. In Algorithm 3, we initialized $\mathbf{x}^0$ as a zero vector, and $\mathbf{x}^1$ was set to the ℓ_1 norm solution.

Figures 4 and 5 illustrate the relationship between average error, sparsity, and μ for various values of n/m. A discernible trend is observed wherein the average error decreases while sparsity increases with an increase in μ. When μ is small, greater emphasis is

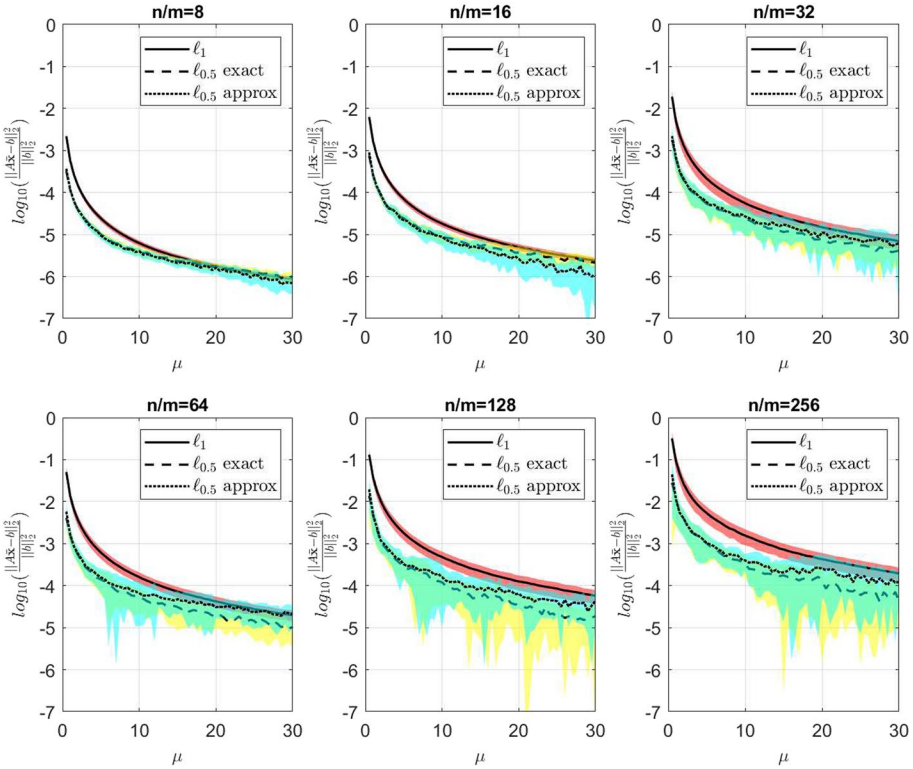


Fig. 4 Average error vs μ for different values of n/m. Yellow and cyan shades are the standard deviations for the exact and approximate $\ell_{0.5}$ quasi-norms, respectively

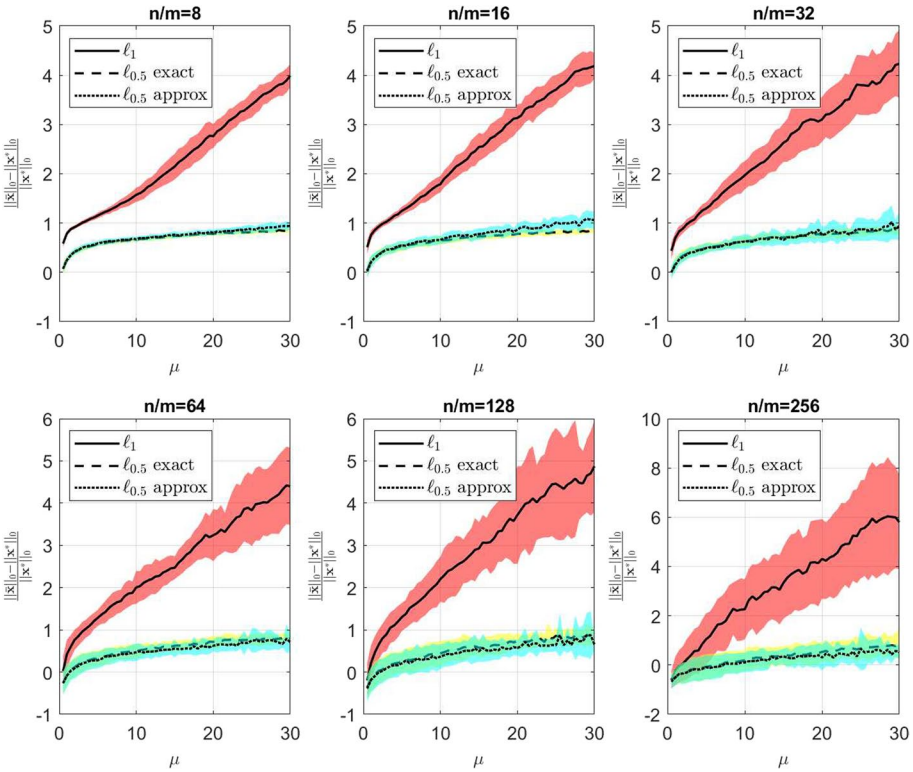


Fig. 5 Sparsity vs μ for different values of n/m. Yellow and cyan shades are the standard deviations for the exact and approximate $\ell_{0.5}$ quasi-norms, respectively

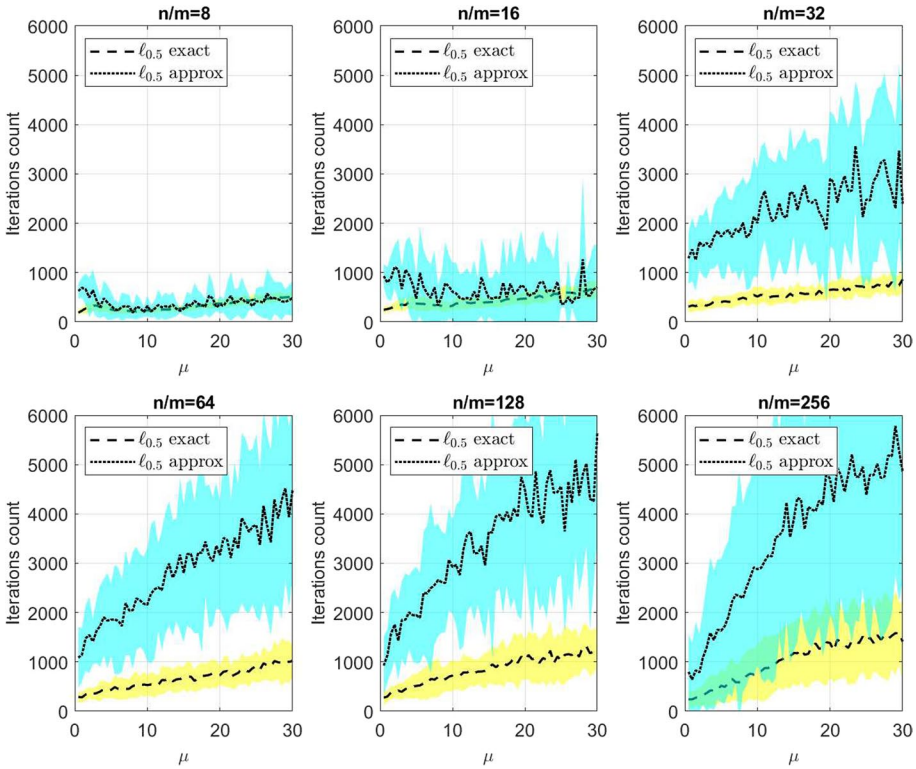


Fig. 6 Iterations count vs μ for different values of n/m

placed on the loss function, emphasizing $\ell_{0.5}$ quasi-norm minimization. Consequently, the sparsity level, as depicted in Fig. 5, remains low. Conversely, for higher values of μ, more weight is assigned to the regularization term's minimization, resolving $\|\mathbf{Ax}-\mathbf{b}\|^2$, resulting in decreased error (Fig. 4) accompanied by an increase in sparsity.

Figure 6 provides insight into the statistics of the number of iterations until convergence for both the $\ell_{0.5}$ exact and approximate algorithms. Notably, with a sufficient number of available realizations, specifically for $n/m = 8$ and $n/m = 16$, both algorithms require approximately the same number of iterations. However, as the number of available realizations decreases, particularly for $n/m = 32$ and higher, the exact proximal solution exhibits a significantly lower number of iterations to converge. This observation, coupled with the findings in Figs. 4 and 5, suggests that our algorithm not only yields a comparable solution to the approximate method but also converges with fewer iterations.

7 Conclusion

In this paper, we introduced a non-convex ADMM algorithm, denoted as pQN-ADMM, designed for solving the ℓ_p quasi-norm minimization problem. Significantly, our proposed algorithm serves as a versatile approach for tackling ℓ_p problems, as it does not rely on specific structural assumptions for the convex constraint set. Moreover, we delved into the problem of solving a non-convex relaxation of RMPs utilizing the Schatten-p quasi-norm. This relaxation was established as the ℓ_p minimization of

the singular values of the variable matrix, rendering it amenable to the pQN-ADMM algorithm. For scenarios involving constraints defined by differentiable functions with Lipschitz continuous gradients, a proximal gradient step was employed, mitigating the need for a convex projection step. This enhancement not only accelerates the algorithm but also ensures its convergence. Illustrating the numerical results, we applied the pQN-ADMM to diverse examples, encompassing sparse vector reconstruction, matrix completion, and system identification. The algorithm demonstrated competitiveness against various ℓ_p-based baselines, underscoring its efficacy across a spectrum of applications.

Acknowledgements
Not applicable.

Author contributions
The problem formulation was initiated by CM. Analysis and development of the non-convex projection algorithms for sparse vector recovery and rank minimization were jointly developed by OS and ME. In section 6, ME conducted the experiments related to sparse vectors, while OS was responsible for the remaining numerical experiments and writing the paper. NS proposed the proximal gradient method and designed the framework for its numerical experiments. The paper was reviewed and edited by CM and NS. All authors read and approved the final manuscript.

Funding
This work has been partially supported by the National Institutes of Health (NIH) Grant R01 HL142732 and National Science Foundation (NSF) Grant 1808266.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
All the authors provide EURASIP the consent for publication.

Competing interests
The authors declare no competing interests.

Received: 17 November 2023 Accepted: 16 January 2024
Published online: 07 February 2024

References

1. L. Vandenberghe, S. Boyd, Semidefinite programming. SIAM Rev. **38**(1), 49–95 (1996). https://doi.org/10.1137/1038003
2. J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
3. E.J. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
4. D.L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
5. A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**(1), 34–81 (2009)
6. J.A. Tropp, S.J. Wright, Computational methods for sparse solution of linear inverse problems. Proc. IEEE **98**(6), 948–958 (2010)
7. S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries. IEEE Trans. Signal Process. **41**(12), 3397–3415 (1993)
8. J.A. Tropp, Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**(10), 2231–2242 (2004)
9. R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization. IEEE Signal Process. Lett. **14**(10), 707–710 (2007)
10. R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing. in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3869–3872 (2008). IEEE
11. A. Miller, *Subset Selection in Regression* (CRC Press, 2002)
12. E.J. Candes, T. Tao, Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
13. R. Saab, R. Chartrand, O. Yilmaz, Stable sparse approximations via nonconvex optimization. in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3885–3888 (2008)

14. R. Chartrand, Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 262–265 (2009)
15. N. Mourad, J.P. Reilly, Minimizing nonconvex functions for sparse vector reconstruction. IEEE Trans. Signal Process. **58**(7), 3485–3496 (2010)
16. R. Chartrand, B. Wohlberg, A nonconvex ADMM algorithm for group sparsity with sparse groups. in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6009–6013 (2013). https://doi.org/10.1109/ICASSP.2013.6638818
17. S. Boyd, N. Parikh, E. Chu, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers* (Now Publishers Inc, 2011)
18. Z. Xu, X. Chang, F. Xu, H. Zhang, $l_{1/2}$ regularization: a thresholding representation theory and a fast solver. IEEE Trans. Neural Netw. Learn. Syst. **23**(7), 1013–1027 (2012). https://doi.org/10.1109/TNNLS.2012.2197412
19. J. Zeng, S. Lin, Y. Wang, Z. Xu, $l_{1/2}$ regularization: convergence of iterative half thresholding algorithm. IEEE Trans. Signal Process. **62**(9), 2317–2329 (2014). https://doi.org/10.1109/TSP.2014.2309076
20. G. Li, T.K. Pong, Global convergence of splitting methods for nonconvex composite optimization. SIAM J. Optim. **25**(4), 2434–2460 (2015)
21. Y. Wang, W. Yin, J. Zeng, Global convergence of ADMM in nonconvex nonsmooth optimization. J. Sci. Comput. **78**(1), 29–63 (2019)
22. J. Zhang, S. Zhang, W. Wang, Robust signal recovery for ℓ_{1-2} minimization via prior support information. Inverse Prob. **37**(11), 115001 (2021). https://doi.org/10.1088/1361-6420/ac274a
23. W. Wang, J. Zhang, Performance guarantees of regularized ℓ_{1-2} minimization for robust sparse recovery. Signal Process. **201**, 108730 (2022)
24. X. Luo, N. Feng, X. Guo, Z. Zhang, Exact recovery of sparse signals with side information. EURASIP J. Adv. Signal Process. **2022**(1), 1–14 (2022)
25. M. Mesbahi, A semi-definite programming solution of the least order dynamic output feedback synthesis problem. in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 2, pp. 1851–18562 (1999). https://doi.org/10.1109/CDC.1999.830903
26. M. Fazel, H. Hindi, S.P. Boyd, A rank minimization heuristic with application to minimum order system approximation. in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 6, pp. 4734–47396 (2001)
27. M. Fazel, H. Hindi, S.P. Boyd, Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. in *Proceedings of the 2003 American Control Conference*, 2003., vol. 3, pp. 2156–21623 (2003)
28. M. Fazel, H. Hindi, S. Boyd, Rank minimization and applications in system theory. in *Proceedings of the 2004 American Control Conference*, vol. 4, pp. 3273–32784 (2004)
29. F. Nie, H. Huang, C. Ding, Low-rank matrix recovery via efficient Schatten p-norm minimization. in *Twenty-sixth AAAI Conference on Artificial Intelligence* (2012)
30. F. Nie, H. Wang, H. Huang, C. Ding, Joint schatten *p*-norm and l_p norm robust matrix completion for missing value recovery. Knowl. Inf. Syst. **42**(3), 525–544 (2015)
31. R. Chartrand, Nonconvex splitting for regularized low-rank + sparse decomposition. IEEE Trans. Signal Process. **60**(11), 5810–5819 (2012). https://doi.org/10.1109/TSP.2012.2208955
32. M.D. Gupta, S. Kumar, Non-convex p-norm projection for robust sparsity. in *2013 IEEE International Conference on Computer Vision*, pp. 1593–1600 (2013). https://doi.org/10.1109/ICCV.2013.201
33. S. Bahmani, B. Raj, A unifying analysis of projected gradient descent for ℓ_p-constrained least squares. Appl. Comput. Harmon. Anal. **34**(3), 366–378 (2013)
34. C. Helmberg, F. Rendl, A spectral bundle method for semidefinite programming. SIAM J. Optim. **10**(3), 673–696 (2000)
35. A. Beck, M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. **31**(3), 167–175 (2003)
36. Y. Nesterov, A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming* (SIAM, 1994)
37. A. Ben-Tal, A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications* (SIAM, 2001)
38. Z. Zha, X. Zhang, Y. Wu, Q. Wang, X. Liu, L. Tang, X. Yuan, Non-convex weighted l_p nuclear norm based ADMM framework for image restoration. Neurocomputing **311**, 209–224 (2018)
39. L. Mirsky, A trace inequality of John von Neumann. Monatshefte für mathematik **79**(4), 303–306 (1975)
40. N. Parikh, S. Boyd, Proximal algorithms. Found. Trends Optim. **1**(3), 127–239 (2014)
41. Q. Yao, J.T. Kwok, F. Gao, W. Chen, T.-Y. Liu, Efficient inexact proximal gradient algorithm for nonconvex problems. arXiv preprint arXiv:1612.09069 (2016)
42. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
43. H. Attouch, J. Bolte, B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. Math. Program. **137**(1), 91–129 (2013)
44. P. Gong, C. Zhang, Z. Lu, J. Huang, J. Ye, A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. in *International Conference on Machine Learning*, pp. 37–45 (2013). PMLR
45. H. Attouch, J. Bolte, P. Redont, A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
46. J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**(1), 459–494 (2014)
47. M. Razaviyayn, M. Hong, Z.-Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. **23**(2), 1126–1153 (2013)
48. P. Tseng, S. Yun, A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**(1), 387–423 (2009)

49. Y. Hu, C. Li, K. Meng, X. Yang, Linear convergence of inexact descent method and inexact proximal gradient algorithms for lower-order regularization problems. J. Global Optim. **79**(4), 853–883 (2021)
50. M. ApS, The MOSEK Optimization Toolbox for MATLAB Manual. Version 9.0. (2019). http://docs.mosek.com/9.0/toolbox/index.html
51. S. Foucart, M.-J. Lai, Sparsest solutions of under-determined linear systems via ℓ_q-minimization for $0 < q \leq 1$. Appl. Comput. Harmon. Anal. **26**(3), 395–407 (2009). https://doi.org/10.1016/j.acha.2008.09.001
52. K. Benidis, Y. Feng, D.P. Palomar, Sparse portfolios for high-dimensional financial index tracking. IEEE Trans. Signal Process. **66**(1), 155–170 (2018). https://doi.org/10.1109/TSP.2017.2762286
53. D. Ge, X. Jiang, Y. Ye, A note on the complexity of ℓ_p minimization. Math. Program. **129**(2), 285–299 (2011)
54. C. Kümmerle, C. Mayrink Verdun, Escaping saddle points in ill-conditioned matrix completion with a scalable second order method. in *Workshop on Beyond First Order Methods in ML Systems at the*37^{th} *International Conference on Machine Learning* (2020)
55. C. Kümmerle, C. Mayrink Verdun, A scalable second order method for ill-conditioned matrix completion from few samples. in *International Conference on Machine Learning (ICML)* (2021)
56. M.-J. Lai, Y. Xu, W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed ℓq minimization. SIAM J. Numer. Anal. **51**(2), 927–957 (2013)
57. K. Mohan, M. Fazel, Iterative reweighted algorithms for matrix rank minimization. J. Mach. Learn. Res. **13**(110), 3441–3473 (2012)
58. K. Mohan, M. Fazel, Reweighted nuclear norm minimization with application to system identification. in *Proceedings of the 2010 American Control Conference*, pp. 2953–2959 (2010)
59. M. Sznaier, M. Ayazoglu, T. Inanc, Fast structured nuclear norm minimization with applications to set membership systems identification. IEEE Trans. Autom. Control **59**(10), 2837–2842 (2014)
60. Z. Liu, L. Vandenberghe, Semidefinite programming methods for system realization and identification. in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, pp. 4676–4681 (2009)
61. M. Fazel, T.K. Pong, D. Sun, P. Tseng, Hankel matrix rank minimization with applications to system identification and realization. SIAM J. Matrix Anal. Appl. **34**(3), 946–977 (2013)
62. N.S. Aybat, G. Iyengar, A first-order augmented Lagrangian method for compressed sensing. SIAM J. Optim. **22**(2), 429–459 (2012)
63. E.T. Hale, W. Yin, Y. Zhang, A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. CAAM TR07-07, Rice University 43, 44 (2007)
64. C. O'Brien, M.D. Plumbley, Inexact proximal operators for ℓ_p-Quasi-norm minimization. in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4724–4728 (2018). https://doi.org/10.1109/ICASSP.2018.8462524

Publisher's Note