

Article

A Designed Thresholding Operator for Low-Rank Matrix Completion

Angang Cui ^{1,*} , Haizhen He ² and Hong Yang ¹ ¹ School of Mathematics and Statistics, Yulin University, Yulin 719000, China; yanghong@yulinu.edu.cn² School of International Education, Yulin University, Yulin 719000, China; hehaizhen@yulinu.edu.cn

* Correspondence: cuiangang@yulinu.edu.cn

Abstract: In this paper, a new thresholding operator, namely, designed thresholding operator, is designed to recover the low-rank matrices. With the change of parameter in designed thresholding operator, the designed thresholding operator can apply less bias to the larger singular values of a matrix compared with the classical soft thresholding operator. Based on the designed thresholding operator, an iterative thresholding algorithm for recovering the low-rank matrices is proposed. Numerical experiments on some image inpainting problems show that the proposed thresholding algorithm performs effectively in recovering the low-rank matrices.

Keywords: low-rank matrix completion; designed thresholding operator; iterative matrix designed thresholding algorithm; image inpainting

MSC: 65K05; 90C26



Citation: Cui, A.; He, H.; Yang, H. A Designed Thresholding Operator for Low-Rank Matrix Completion. *Mathematics* **2024**, *12*, 1065. <https://doi.org/10.3390/math12071065>

Academic Editor: Ioannis K. Argyros

Received: 16 February 2024

Revised: 29 March 2024

Accepted: 30 March 2024

Published: 2 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the problem of recovering an unknown low-rank matrix from its limited number of observed entries has been actively studied in many scientific applications such as Netflix problem [1], image processing [2], system identification [3], video denoising [4], signal processing [5], subspace learning [6], and so on. In mathematics, this problem can be modeled by the following low-rank matrix completion problem:

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \text{rank}(\mathbf{X}) \text{ s.t. } \mathbf{X}_{i,j} = \mathbf{M}_{i,j}, \forall (i,j) \in \Omega, \quad (1)$$

where $\Omega \subset \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$ is the set of indices of observed entries. Without loss of generality, throughout this paper, we assume that $m \leq n$. If we summarize the observed entries via $\mathcal{P}_\Omega(\mathbf{M})$, where the projection $\mathcal{P}_\Omega : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ is defined by

$$[\mathcal{P}_\Omega(\mathbf{X})]_{i,j} = \begin{cases} \mathbf{X}_{i,j}, & (i,j) \in \Omega; \\ 0, & \text{otherwise,} \end{cases}$$

the low-rank matrix completion problem (1) can be rewritten as

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \text{rank}(\mathbf{X}) \text{ s.t. } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}). \quad (2)$$

Unfortunately, the problem (2) is NP-hard and all known algorithms for exactly solving it are time doubly exponential in both theory and practice [1,7]. To overcome such a difficulty, many researchers (e.g., [1,3,5,8–10]) have suggested to relax the rank function $\text{rank}(\mathbf{X})$ by the nuclear norm $\|\mathbf{X}\|_*$, which leads to the following nuclear norm minimization problem:

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \|\mathbf{X}\|_* \text{ s.t. } \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M}), \quad (3)$$

where $\|\mathbf{X}\|_* = \sum_{i=1}^m \sigma_i(\mathbf{X})$ denotes the nuclear norm of matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, and $\sigma_i(\mathbf{X})$, $i = 1, \dots, m$, are the singular values of $\mathbf{X} \in \mathbb{R}^{m \times n}$.

The nuclear norm $\|\mathbf{X}\|_*$, as a convex relaxation of the rank function $\text{rank}(\mathbf{X})$, can be considered as the best convex approximation of the rank function $\text{rank}(\mathbf{X})$. In theory, Candès et al. [1,11] have proved that, if the observed entries are selected uniformly at random, the unknown low-rank matrix can be exactly recovered with high probability by solving the problem (2). As a convex optimization problem, the problem (2) can be solved by using the semidefinite programming solvers such as SeDuMi [12] and SDPT3 [13]. However, these algorithms are unsuitable to handle the large size problems due to the high computational cost and the memory requirements. Different from the semidefinite programming solvers, the regularized version of the problem (2), i.e.,

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{M})\|_F^2 + \lambda \|\mathbf{X}\|_* \right\} \quad (4)$$

has been frequently studied in the literature to solve the large-scale matrix completion, where $\lambda > 0$ is a regularization parameter representing the tradeoff between error and nuclear norm. There are many algorithms proposed to solve the problem (3). The most popular algorithms include the singular value thresholding (SVT) algorithm [8], accelerated proximal gradient (APG) algorithm [14], linearized augmented lagrangian and alternating direction methods [15], and so on. These algorithms are all based on the soft thresholding operator [16] to recover the unknown low-rank matrices. Although the problem (4) possesses many algorithmic advantages, these iterative soft thresholding version algorithms tend to have the bias on shrinking all the singular values of matrices, and sometimes result in over-penalization as the l_1 -norm in compressed sensing.

To reduce the bias, in this paper we design a new thresholding operator, namely, designed thresholding operator, to recover the low-rank matrices, which has less bias than the soft thresholding operator. The numerical experiments on some low-rank matrix completion problems show that the proposed thresholding operator performs efficiently in recovering the low-rank matrices.

This paper is organized as follows. In Section 2, we give the definition of the designed thresholding operator, and study some properties for the designed thresholding operator. In Section 3, an iterative thresholding algorithm based on the designed thresholding operator is generated to recover the low-rank matrices. In Section 4, some numerical experiments are presented to verify the effectiveness of the proposed algorithm. Finally, we draw some conclusions in Section 5.

2. Designed Thresholding Operator

In this section, we first review the classical soft thresholding operator [16], and then design a new thresholding operator, namely, designed thresholding operator, to recover the low-rank matrices.

2.1. Soft Thresholding Operator

For any fixed $\lambda > 0$ and $t \in \mathbb{R}$, the soft thresholding operator [16] is defined as

$$s_\lambda(t) = \max\{|t| - \lambda, 0\} \text{sign}(t), \quad (5)$$

which can be explained by the fact that it is the proximity operator of $|x|$, i.e.,

$$s_\lambda(t) := \arg \min_{x \in \mathbb{R}} \left\{ \frac{1}{2} (x - t)^2 + \lambda |x| \right\}. \quad (6)$$

The soft thresholding operator plays a significant role in solving problem (4). However, it has the biased estimation and sometimes results in over-penalization. The behavior of the soft thresholding operator is plotted in Figure 1, and we can see that the bias between t and soft thresholding operator $s_\lambda(t)$ equals to λ .

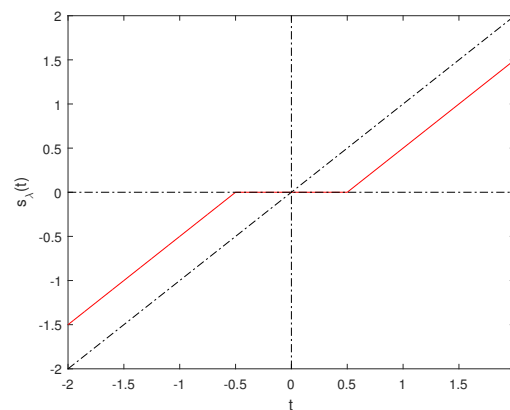


Figure 1. Plot of the soft thresholding operator $s_\lambda(t)$ with $\lambda = 0.5$.

2.2. Designed Thresholding Operator

To reduce the bias, in this subsection we design a new thresholding operator, namely, designed thresholding operator, to recover the low-rank matrices.

Definition 1. (Designed thresholding operator) For any fixed $\lambda > 0$, $c \geq 0$ and $t \in \mathbb{R}$, the designed thresholding operator is defined as

$$r_{c,\lambda}(t) = \max \left\{ |t| - \frac{\lambda(c + \lambda)}{c + |t|}, 0 \right\} \text{sign}(t). \quad (7)$$

According to Definition 1, we immediately obtain the following Property 1, which shows that the designed thresholding operator has less bias to the larger coefficients than the soft thresholding operator.

Property 1. For any $c \in [0, +\infty)$, the bias $t - r_{c,\lambda}(t)$ approaches zero as the magnitude of t increases.

Proof. Without loss of generality, we assume $t \geq 0$. Then, we have $t - r_{c,\lambda}(t) \leq \lambda$ for any $t \geq 0$, and $t - r_{c,\lambda}(t) = \frac{\lambda(c + \lambda)}{c + t}$ for any $t \geq \lambda$. It is easy to verify that $t - r_{c,\lambda}(t) = \frac{\lambda(c + \lambda)}{c + t} \rightarrow 0$ if $t \rightarrow +\infty$. The proof is thus complete. $\square$

In addition, we can also observe that the designed thresholding operator approximates soft thresholding operator when $c \rightarrow +\infty$, i.e.,

$$\lim_{c \rightarrow +\infty} r_{c,\lambda} = \max \{ |t| - \lambda, 0 \} \text{sign}(t).$$

The behaviors of the designed thresholding operator for some $c \geq 0$ with $\lambda = 0.5$ are plotted in Figure 2. With the change of parameter $c > 0$, the designed thresholding operator can apply less bias to the larger coefficients.

According to ([17], Theorem 1), we immediately obtain the following Lemma 1, which shows that the designed thresholding operator is in fact the proximal mapping of a non-convex penalty function.

Lemma 1. The designed threshold operator $r_{c,\lambda}(t)$ is the proximal mapping of a penalty function $g(t)$, i.e.,

$$r_{c,\lambda}(t) = \arg \min_{x \in \mathbb{R}} \left\{ \frac{1}{2}(x - t)^2 + \lambda g(x) \right\}, \quad (8)$$

where g is even, nondecreasing and continuous on $[0, +\infty)$, differentiable on $(0, +\infty)$, and non-differentiable at 0 with $\partial g(0) = [-1, 1]$. Moreover, g is concave on $[0, +\infty)$ and satisfies the triangle inequality.

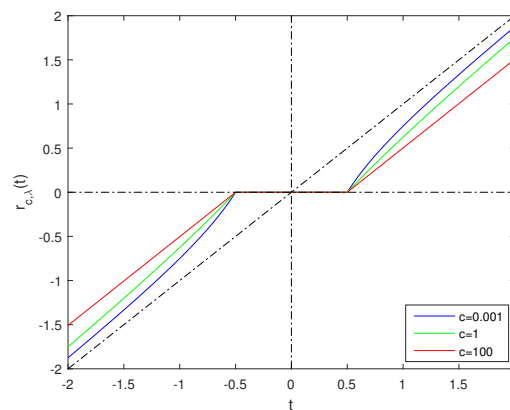


Figure 2. Plot of the designed thresholding operator $r_{c,\lambda}(t)$ for some c with $\lambda = 0.5$.

Definition 2. (Matrix designed thresholding operator) Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, let $\mathbf{X} = \mathbf{U}[\text{Diag}(\{\sigma_i(\mathbf{X})\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}]\mathbf{V}^\top$ be the singular value decomposition (SVD) of matrix $\mathbf{X}$, where $\mathbf{U}$ is a $m \times m$ unitary matrix, $\mathbf{V}$ is a $n \times n$ unitary matrix, $\sigma_i(\mathbf{X})$ is the i -th largest singular value of $\mathbf{X}$, $\text{Diag}(\{\sigma_i(\mathbf{X})\}_{1 \leq i \leq m}) \in \mathbb{R}^{m \times m}$ is the diagonal matrix of the singular values of matrix $\mathbf{X}$ arranged in descending order $\sigma_1(\mathbf{X}) \geq \sigma_2(\mathbf{X}) \geq \dots \geq \sigma_m(\mathbf{X}) \geq 0$, and $\mathbf{0}_{m \times (n-m)}$ is a $m \times (n-m)$ zero matrix. For any fixed $\lambda > 0$ and $c \geq 0$, the matrix designed thresholding operator $\mathcal{R}_{c,\lambda}(\mathbf{X})$ is defined as

$$\mathcal{R}_{c,\lambda}(\mathbf{X}) = \mathbf{U}[\text{Diag}(\{r_{c,\lambda}(\sigma_i(\mathbf{X}))\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}]\mathbf{V}^\top, \quad (9)$$

where $r_{c,\lambda}(\cdot)$ is defined in Definition 1.

For any matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, define the function $G : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}_+$ as

$$G(\mathbf{X}) = \sum_{i=1}^m g(\sigma_i(\mathbf{X})), \quad (10)$$

where $\mathbb{R}_+ := [0, +\infty)$, we can get the following result.

Theorem 1. For any fixed $\lambda > 0$ and $\mathbf{Y} \in \mathbb{R}^{m \times n}$, suppose that $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$ is the optimal solution of the problem

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} \|\mathbf{X} - \mathbf{Y}\|_F^2 + \lambda G(\mathbf{X}) \right\}. \quad (11)$$

Then $\hat{\mathbf{X}}$ can be expressed as

$$\hat{\mathbf{X}} = \mathcal{R}_{c,\lambda}(\mathbf{Y}). \quad (12)$$

Before we give the proof of Theorem 1, we need prepare the following Lemma 2 which plays the key rule for proving Theorem 1.

Lemma 2. (von Neumann's trace inequality) For any matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m \times n}$,

$$\text{Tr}(\mathbf{X}^T \mathbf{Y}) \leq \sum_{i=1}^m \sigma_i(\mathbf{X}) \sigma_i(\mathbf{Y}),$$

where $\sigma_1(\mathbf{X}) \geq \sigma_2(\mathbf{X}) \geq \dots \geq \sigma_m(\mathbf{X})$ and $\sigma_1(\mathbf{Y}) \geq \sigma_2(\mathbf{Y}) \geq \dots \geq \sigma_m(\mathbf{Y})$ are the singular values of $\mathbf{X}$ and $\mathbf{Y}$, respectively. The equality holds if and only if there exists unitary matrices $\mathbf{U}$ and $\mathbf{V}$ such that

$$\mathbf{X} = \mathbf{U}[\text{Diag}(\{\sigma_i(\mathbf{X})\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}]\mathbf{V}^T$$

and

$$\mathbf{Y} = \mathbf{U}[\text{Diag}(\{\sigma_i(\mathbf{Y})\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}] \mathbf{V}^T$$

are the SVDs of $\mathbf{X}$ and $\mathbf{Y}$, simultaneously.

Now, we give the proof of Theorem 1.

Proof. (of Theorem 1) By Lemma 2, we have

$$\begin{aligned} \|\mathbf{X} - \mathbf{Y}\|_F^2 &= \text{Tr}(\mathbf{X}^T \mathbf{X}) - 2\text{Tr}(\mathbf{X}^T \mathbf{Y}) + \text{Tr}(\mathbf{Y}^T \mathbf{Y}) \\ &= \sum_{i=1}^m \sigma_i^2(\mathbf{X}) - 2\text{Tr}(\mathbf{X}^T \mathbf{Y}) + \sum_{i=1}^m \sigma_i^2(\mathbf{Y}) \\ &\geq \sum_{i=1}^m \sigma_i^2(\mathbf{X}) - 2 \sum_{i=1}^m \sigma_i(\mathbf{X}) \sigma_i(\mathbf{Y}) + \sum_{i=1}^m \sigma_i^2(\mathbf{Y}) \\ &= \sum_{i=1}^m (\sigma_i(\mathbf{X}) - \sigma_i(\mathbf{Y}))^2. \end{aligned}$$

Notice that the equality holds if and only if $\mathbf{X}$ and $\mathbf{Y}$ share the same left and right unitary matrices, we assume that

$$\mathbf{X} = \mathbf{U}[\text{Diag}(\{\sigma_i(\mathbf{X})\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}] \mathbf{V}^T$$

and

$$\mathbf{Y} = \mathbf{U}[\text{Diag}(\{\sigma_i(\mathbf{Y})\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}] \mathbf{V}^T$$

are the SVDs of $\mathbf{X}$ and $\mathbf{Y}$, simultaneously. Therefore, it holds that

$$\|\mathbf{X} - \mathbf{Y}\|_F^2 = \sum_{i=1}^m (\sigma_i(\mathbf{X}) - \sigma_i(\mathbf{Y}))^2,$$

and the problem (11) reduces to

$$\min_{\sigma_1(\mathbf{X}) \geq \sigma_2(\mathbf{X}) \geq \dots \geq \sigma_m(\mathbf{X}) \geq 0} \sum_{i=1}^m \left\{ \frac{1}{2} (\sigma_i(\mathbf{X}) - \sigma_i(\mathbf{Y}))^2 + \lambda g(\sigma_i(\mathbf{X})) \right\}. \quad (13)$$

The objective function in (13) is separable, hence, solving the problem (13) is equivalent to solving the following n problems, for each $i = 1, 2, \dots, n$,

$$\min_{\sigma_i(\mathbf{X}) \geq 0} \left\{ \frac{1}{2} (\sigma_i(\mathbf{X}) - \sigma_i(\mathbf{Y}))^2 + \lambda g(\sigma_i(\mathbf{X})) \right\}. \quad (14)$$

Let $\hat{\sigma}_i(\mathbf{X})$ be the optimal solution of the problem (14), by Lemma 1, $\hat{\sigma}_i(\mathbf{X})$ can be expressed as

$$\hat{\sigma}_i(\mathbf{X}) = r_{c,\lambda}(\sigma_i(\mathbf{Y})). \quad (15)$$

Therefore, we can get the optimal solution $\hat{\mathbf{X}}$ of the problem (11) as follows

$$\hat{\mathbf{X}} = \mathbf{U}[\text{Diag}(\{r_{c,\lambda}(\sigma_i(\mathbf{Y}))\}_{1 \leq i \leq m}), \mathbf{0}_{m \times (n-m)}] \mathbf{V}^T, \quad (16)$$

namely,

$$\hat{\mathbf{X}} = \mathcal{R}_{c,\lambda}(\mathbf{Y}).$$

This completes the proof. $\square$

3. Iterative Matrix Designed Thresholding Algorithm

In this section, we present an iterative thresholding algorithm to recover the low-rank matrices using the proposed designed thresholding operator.

Consider the following minimization problem:

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{M})\|_F^2 + \lambda G(\mathbf{X}) \right\}, \quad (17)$$

we can verify the optimal solution of problem (17) can be analytically expressed by the designed thresholding operator.

For any fixed $\lambda > 0$, $\mu > 0$ and $\mathbf{Z} \in \mathbb{R}^{m \times n}$, let

$$C_\lambda(\mathbf{X}) := \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{M})\|_F^2 + \lambda G(\mathbf{X}), \quad (18)$$

$$C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z}) := \mu \left[C_\lambda(\mathbf{X}) - \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Z})\|_F^2 \right] + \frac{1}{2} \|\mathbf{X} - \mathbf{Z}\|_F^2 \quad (19)$$

and

$$B_\mu(\mathbf{Z}) := \mathbf{Z} + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{Z})).$$

The function $C_\lambda(\mathbf{X})$ defined in (18) is the objective function of the problem (17), and the function $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$ defined in (19) is a surrogate function of $C_\lambda(\mathbf{X})$. Clearly, $C_{\lambda,\mu}(\mathbf{X}, \mathbf{X}) = \mu C_\lambda(\mathbf{X})$. By (19), we would expect minimizing the problem (17) by minimizing the surrogate function $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$.

Lemma 3. For any fixed $\lambda > 0$, $\mu > 0$ and $\mathbf{Z} \in \mathbb{R}^{m \times n}$, if $\mathbf{X}^\sharp \in \mathbb{R}^{m \times n}$ is the minimizer of $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$ on $\mathbf{X} \in \mathbb{R}^{m \times n}$, then $\mathbf{X}^\sharp$ satisfies

$$\mathbf{X}^\sharp = \mathcal{R}_{c,\lambda\mu}(B_\mu(\mathbf{Z})). \quad (20)$$

Proof. By definition, the function $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$ can be rewritten as

$$\begin{aligned} C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z}) &= \frac{1}{2} \|\mathbf{X} - (\mathbf{Z} + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{Z})))\|_F^2 + \lambda\mu G(\mathbf{X}) + \frac{\mu}{2} \|\mathcal{P}_\Omega(\mathbf{M})\|_F^2 \\ &\quad + \frac{1}{2} \|\mathbf{Z}\|_F^2 - \frac{\mu}{2} \|\mathcal{P}_\Omega(\mathbf{Z})\|_F^2 - \frac{1}{2} \|\mathbf{Z} + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{Z}))\|_F^2 \\ &= \frac{1}{2} \|\mathbf{X} - B_\mu(\mathbf{Z})\|_F^2 + \lambda\mu G(\mathbf{X}) + \frac{\mu}{2} \|\mathcal{P}_\Omega(\mathbf{M})\|_F^2 + \frac{1}{2} \|\mathbf{Z}\|_F^2 \\ &\quad - \frac{\mu}{2} \|\mathcal{P}_\Omega(\mathbf{Z})\|_F^2 - \frac{1}{2} \|B_\mu(\mathbf{Z})\|_F^2. \end{aligned}$$

This means that, for any fixed $\lambda > 0$, $\mu > 0$ and $\mathbf{Z} \in \mathbb{R}^{m \times n}$, minimizing the function $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$ on $\mathbf{X} \in \mathbb{R}^{m \times n}$ is equivalent to solving the following minimization problem

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2} \|\mathbf{X} - B_\mu(\mathbf{Z})\|_F^2 + \lambda\mu G(\mathbf{X}) \right\}. \quad (21)$$

By Theorem 1, the minimizer $\mathbf{X}^\sharp$ of $C_{\lambda,\mu}(\mathbf{X}, \mathbf{Z})$ on $\mathbf{X} \in \mathbb{R}^{m \times n}$ can be expressed as

$$\mathbf{X}^\sharp = \mathcal{R}_{c,\lambda\mu}(B_\mu(\mathbf{Z})),$$

which completes the proof. $\square$

Lemma 4. For any fixed $\lambda > 0$ and $0 < \mu \leq 1$. If $\mathbf{X}^* \in \mathbb{R}^{m \times n}$ is the optimal solution of the problem (17), then $\mathbf{X}^*$ also solves the minimization problem

$$\min_{\mathbf{X} \in \mathbb{R}^{m \times n}} C_{\lambda,\mu}(\mathbf{X}, \mathbf{X}^*), \quad (22)$$

that is, for any $\mathbf{X} \in \mathbb{R}^{m \times n}$,

$$C_{\lambda,\mu}(\mathbf{X}^*, \mathbf{X}^*) \leq C_{\lambda,\mu}(\mathbf{X}, \mathbf{X}^*). \quad (23)$$

Proof. Since $0 < \mu \leq 1$, we have

$$\|\mathbf{X} - \mathbf{Z}\|_F^2 - \mu \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Z})\|_F^2 \geq 0.$$

Therefore, we can get that

$$\begin{aligned} C_{\lambda,\mu}(\mathbf{X}, \mathbf{X}^*) &= \mu \left[C_\lambda(\mathbf{X}) - \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Z})\|_F^2 \right] + \frac{1}{2} \|\mathbf{X} - \mathbf{Z}\|_F^2 \\ &\geq \mu C_\lambda(\mathbf{X}) - \frac{\mu}{2} \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Z})\|_F^2 + \frac{1}{2} \|\mathbf{X} - \mathbf{Z}\|_F^2 \\ &\geq \mu C_\lambda(\mathbf{X}) \\ &\geq \mu C_\lambda(\mathbf{X}^*) \\ &= C_{\lambda,\mu}(\mathbf{X}^*, \mathbf{X}^*), \end{aligned}$$

which completes the proof. $\square$

By Lemmas 3 and 4, we can derive that the problem (17) permits a thresholding representation theory for its optimal solution.

Theorem 2. For any fixed $\lambda > 0$ and $0 < \mu \leq 1$. If $\mathbf{X}^* \in \mathbb{R}^{m \times n}$ is the optimal solution of the problem (17), then $\mathbf{X}^*$ can be analytically expressed as

$$\mathbf{X}^* = \mathcal{R}_{c,\lambda\mu}(B_\mu(\mathbf{X}^*)). \quad (24)$$

With representation (24), an algorithm for solving the problem (17) can be naturally given by

$$\mathbf{X}^{k+1} = \mathcal{S}_{\lambda\mu}(B_\mu(\mathbf{X}^k)), \quad (25)$$

where $B_\mu(\mathbf{X}^k) = \mathbf{X}^k + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{X}^k))$. In this paper, we call the iteration (25) the iterative matrix designed thresholding (IMDT) algorithm, which is summarized in Algorithm 1.

Algorithm 1 Iterative matrix designed thresholding (IMDT) algorithm

Input: $\lambda > 0, \mu \in (0, 1], c \geq 0$;
Initialize: $\mathbf{X}^0 \in \mathbb{R}^{m \times n}, k = 0$;
while not converged, **do**
 $B_\mu(\mathbf{X}^k) = \mathbf{X}^k + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{X}^k));$
 $\mathbf{X}^{k+1} = \mathcal{R}_{c,\lambda\mu}(B_\mu(\mathbf{X}^k));$
 $k \rightarrow k + 1$;
end while
return: $\mathbf{X}^{opt}$

Similar to ([18], Theorem 4.1), we can immediately derive the convergence of the IMDT algorithm.

Theorem 3. Let $\{\mathbf{X}^k\}$ be the sequence generated by the IMDT algorithm with $0 < \mu < 1$. Then

- (1) The sequence $\{C_\lambda(\mathbf{X}^k)\}$ is monotonically decreasing and converges to $C_\lambda(\mathbf{X}^*)$, where $\mathbf{X}^*$ is any accumulation point of $\{\mathbf{X}^k\}$.
- (2) The sequence $\{\mathbf{X}^k\}$ is asymptotically regular, i.e., $\lim_{k \rightarrow \infty} \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 = 0$.
- (3) Any accumulation point of the sequence $\{\mathbf{X}^k\}$ is a stationary point of the problem (17).

It is worth mentioning that the quality of the optimal solution generated by IMDT algorithm depends seriously on the setting of the regularization parameter λ . But the selection of optimal regularization parameter λ is a very hard problem. There is no optimal rule for the selection of the optimal regularization parameter λ in general. However, when

some prior information (e.g., rank) is known for the optimal solution of problem (17), it is realistic to set the regularization parameter more reasonably. Following [18,19], we can give a useful parameter-setting rule to select the optimal regularization parameter λ for the IMDT algorithm. The details of the parameter-setting rule can be described as follows.

Suppose that the optimal solution $\mathbf{X}^*$ of the problem (17) is rank r , and the singular values of matrix $B_\mu(\mathbf{X}^*)$ are arranged as

$$\sigma_1(B_\mu(\mathbf{X}^*)) \geq \sigma_2(B_\mu(\mathbf{X}^*)) \geq \cdots \geq \sigma_m(B_\mu(\mathbf{X}^*)) \geq 0.$$

By Theorem 2, we have

$$\begin{aligned}\sigma_i(B_\mu(\mathbf{X}^*)) &> \lambda\mu \Leftrightarrow i \in \{1, 2, \dots, r\}, \\ \sigma_i(B_\mu(\mathbf{X}^*)) &\leq \lambda\mu \Leftrightarrow i \in \{r+1, r+2, \dots, m\},\end{aligned}$$

which implies

$$\frac{\sigma_{r+1}(B_\mu(\mathbf{X}^*))}{\mu} \leq \lambda < \frac{\sigma_r(B_\mu(\mathbf{X}^*))}{\mu}. \quad (26)$$

Estimation (26) can help to set the optimal regularization parameter λ for IMDT algorithm, and a reliable selection can be set as

$$\lambda = \frac{\sigma_{r+1}(B_\mu(\mathbf{X}^*))}{\mu}. \quad (27)$$

In practice, we approximate the unknown real optimal solution $\mathbf{X}^*$ by $\mathbf{X}^k$, that is, we can take

$$\lambda = \lambda_k = \frac{\sigma_{r+1}(B_\mu(\mathbf{X}^k))}{\mu} \quad (28)$$

in each iteration of IMDT algorithm.

When we set the optimal regularization parameter λ using Equation (28), the IMDT algorithm will adapt to select the optimal regularization parameter λ . In this paper, we call IMDT algorithm with parameter-setting rule (28) the adaptive iterative matrix designed thresholding (AIMDT) algorithm which is summarized in Algorithm 2.

Algorithm 2 Adaptive iterative matrix designed thresholding (AIMDT) algorithm

Input: $\lambda > 0, \mu \in (0, 1)$;
Initialize: $\mathbf{X}^0 \in \mathbb{R}^{m \times n}, k = 0$;
while not converged, **do**
 $B_\mu(\mathbf{X}^k) = \mathbf{X}^k + \mu(\mathcal{P}_\Omega(\mathbf{M}) - \mathcal{P}_\Omega(\mathbf{X}^k))$;
 $\lambda_k = \frac{\sigma_{r+1}(B_\mu(\mathbf{X}^k))}{\mu}$;
 $\lambda = \lambda_k$;
 $\mathbf{X}^{k+1} = \mathcal{R}_{c, \lambda\mu}(B_\mu(\mathbf{X}^k))$;
 $k \rightarrow k + 1$;
end while
return: $\mathbf{X}^{opt}$

4. Numerical Experiments

In this section, we only carry out the numerical experiments to test the performance of the AIMDT algorithm on some grayscale image inpainting problems, and then compare it with the classical SVT algorithm [8]. The freedom ration is defined as $\text{FR} = s / (r(m + n - r))$, where s is the cardinality of observation set Ω and r is the rank of the matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. For

$FR < 1$, it is impossible to recover the original low-rank matrix [9]. The stopping criterion of the algorithms is defined as

$$\frac{\|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F}{\max\{\|\mathbf{X}^k\|_F, 1\}} \leq 10^{-8}.$$

Given the original low-rank matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, the accuracy of the generated solution $\mathbf{X}^{opt}$ of the algorithms is measured by the relative error (RE), defined as

$$RE = \frac{\|\mathbf{X}^{opt} - \mathbf{M}\|_F}{\|\mathbf{M}\|_F}.$$

In all numerical experiments, we set $\mu = 0.99$ in AIMDT algorithm. The numerical experiments are all conducted on a personal computer with Matlab platform.

In numerical experiments, the algorithms are tested on three 512×512 grayscale images (Lena, Boat and Fingerprint). It is well known that grayscale images can be expressed by matrices. In grayscale image inpainting, the grayscale value of the pixels of the image are missing, and we want to fill in these missing pixels. If the image is of low-rank, or of numerical low-rank, we can solve the image inpainting problem as a matrix completion problem (2). We first use the SVD to obtain their approximated low-rank images with rank 50. The original images and their corresponding approximated low-rank images are displayed in Figures 3, 4 and 5, respectively. Then, we mask some pixels of these three low-rank images. The 3.81% of elements of image are set to mask, and the masked images are displayed in Figure 6.



Figure 3. Original 512×512 Lena image and its approximated low-rank image with rank 50. (a) Original Lena image. (b) Low-rank Lena image with rank 50.

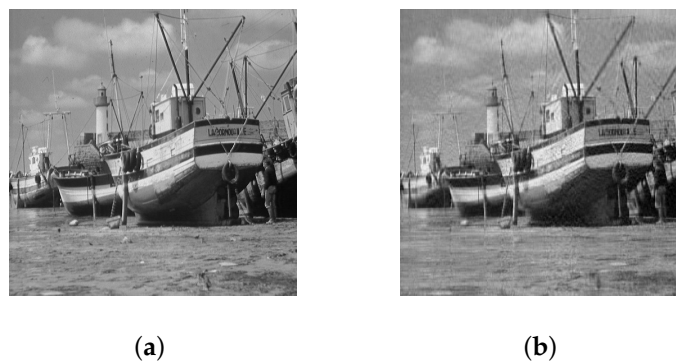


Figure 4. Original 512×512 Boat image and its approximated low-rank image with rank 50. (a) Original Boat image. (b) Low-rank Boat image with rank 50.

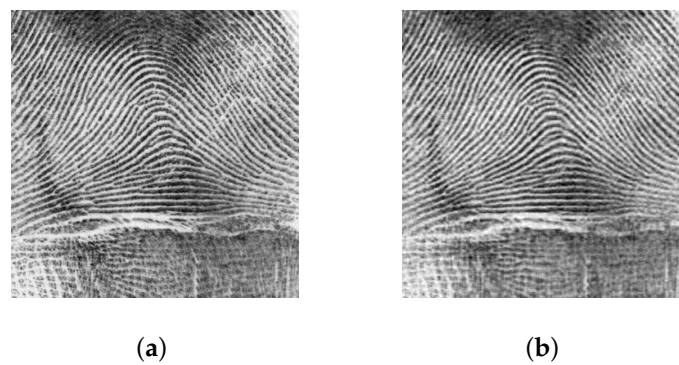


Figure 5. Original 512×512 Fingerprint image and its approximated low-rank image with rank 50. (a) Original Fingerprint image. (b) Low-rank Fingerprint image with rank 50.

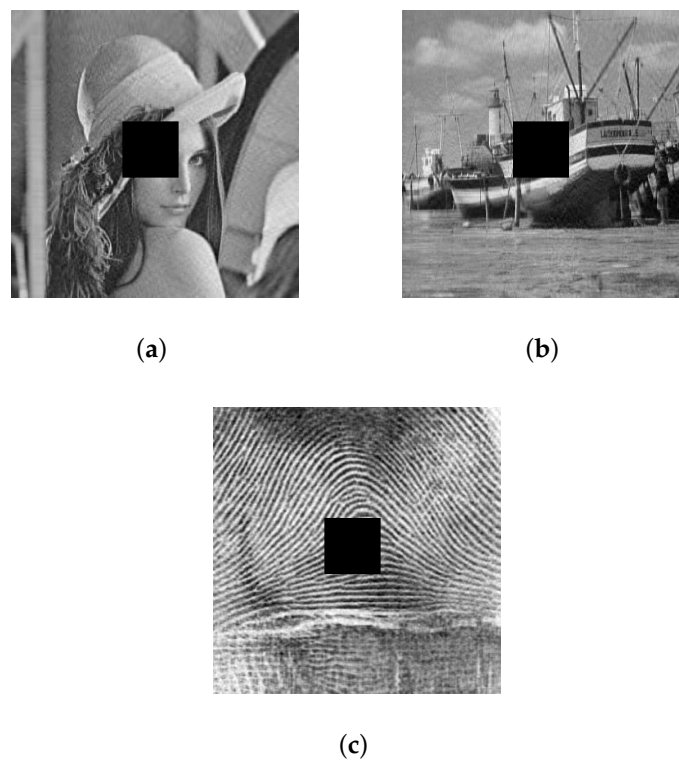


Figure 6. Low-rank images with 3.81% pixels missing. (a) 3.81% pixels missing Lena image. (b) 3.81% pixels missing Boat image. (c) 3.81% pixels missing Fingerprint image.

Table 1 reports the numerical results of the algorithms for image inpainting problems. For AIMDT algorithm, three different c values are considered, that is, $c = 1, 5, 100$. From Table 1, we can make the following observations: (i) AIMDT algorithms with parameter $c = 1, 5, 100$ are all superior to SVT algorithm in RE. In particular, the REs generated by the AIMDT algorithms decreases as the parameter c decreases; (ii) AIMDT algorithms with parameter $c = 1, 5, 100$ are all more time-consuming than SVT algorithm. The calculation time of AIMDT algorithm increases with the decrease of parameter c . In a word, AIMDT algorithms can more accurately recover the low-rank grayscale images than SVT algorithm, and the AIMDT algorithm with parameter $c = 1$ performs best. We display the recovered low-rank Lena, Boat and Fingerprint images in Figures 7, 8 and 9, respectively. From these figures, we can see that the AIMDT algorithm with parameter $c = 1$ has the best performance in recovering the low-rank grayscale images.



Figure 7. Comparisons of algorithms for recovering the low-rank Lena image with 3.81% pixels missing. (a) AIMDT algorithm, $c = 1$. (b) AIMDT algorithm, $c = 5$. (c) AIMDT algorithm, $c = 100$. (d) SVT algorithm.

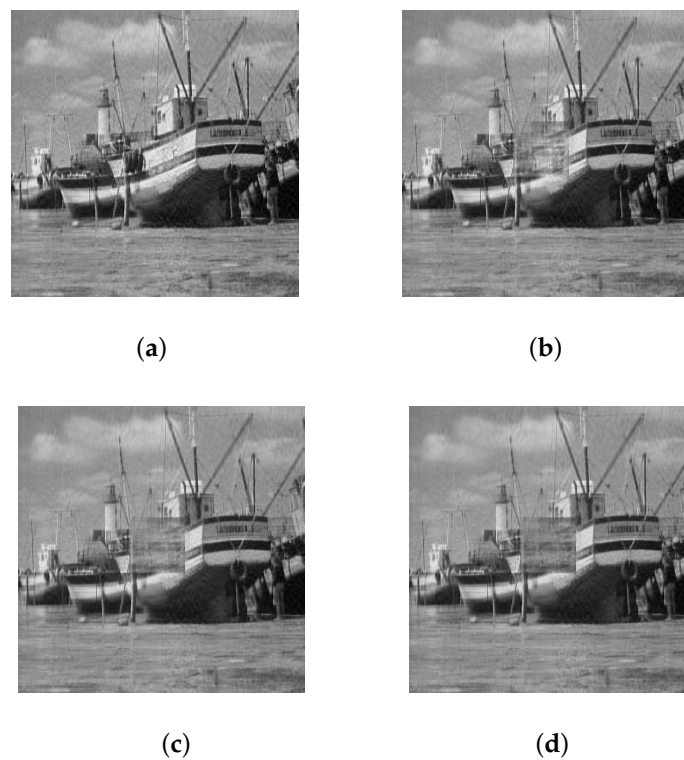


Figure 8. Comparisons of algorithms for recovering the low-rank Boat image with 3.81% pixels missing. (a) AIMDT algorithm, $c = 1$. (b) AIMDT algorithm, $c = 5$. (c) AIMDT algorithm, $c = 100$. (d) SVT algorithm.

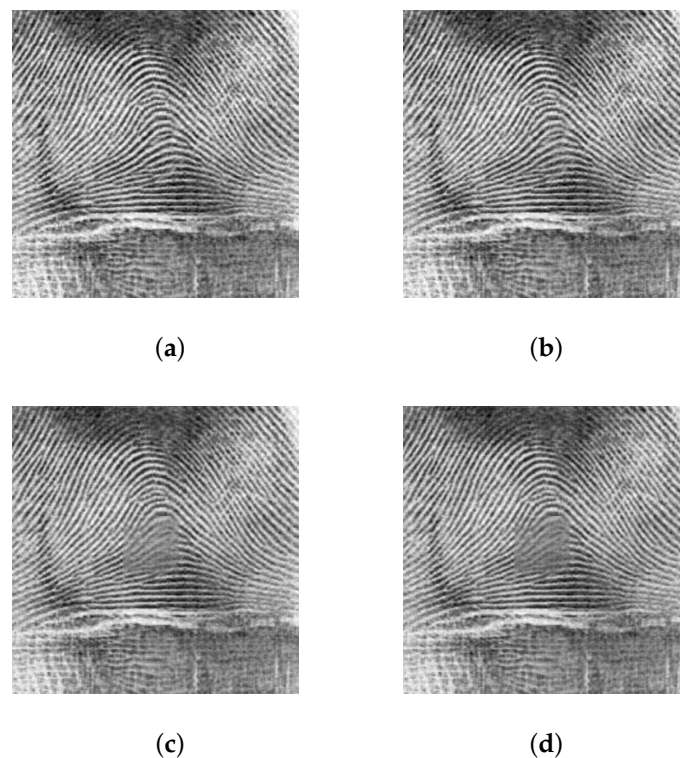


Figure 9. Comparisons of algorithms for recovering the low-rank Fingerprint image with 3.81% pixels missing. (a) AIMDT algorithm, $c = 1$. (b) AIMDT algorithm, $c = 5$. (c) AIMDT algorithm, $c = 100$. (d) SVT algorithm.

Table 1. Numerical results of algorithms for image inpainting problems.

Image (Name, Rank, FR)	AIMDT, $c = 1$		AIMDT, $c = 5$		AIMDT, $c = 100$		SVT Algorithm	
	RE	Time	RE	Time	RE	Time	RE	Time
(Lena, 50, 5.17)	2.11×10^{-2}	226.32	5.71×10^{-2}	76.36	6.67×10^{-2}	28.45	7.29×10^{-2}	21.54
(Boat, 50, 5.17)	3.28×10^{-5}	773.77	6.23×10^{-2}	107.11	8.18×10^{-2}	19.54	8.84×10^{-2}	15.30
(Fingerprint, 50, 5.17)	4.21×10^{-6}	94.00	1.34×10^{-2}	67.86	5.68×10^{-2}	10.03	6.29×10^{-2}	8.03

5. Conclusions

In this paper, based on the designed thresholding operator which has less bias to the larger singular values than the soft thresholding operator, an iterative matrix designed thresholding algorithm and its adaptive version are proposed to recover the low-rank matrices. Numerical experiments on some image inpainting problems show that the adaptive iterative matrix designed thresholding algorithm performs efficiently in recovering the low-rank matrices. In the future work, we shall study the acceleration techniques to improve the convergence rate of the proposed algorithm and explore other applications for the designed thresholding operator.

Author Contributions: Conceptualization, methodology, writing—original draft, writing—review and editing, funding acquisition, A.C.; writing—original draft, writing—review and editing, H.H.; methodology, funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Doctoral Research Project of Yulin University under Grant 21GK04 and Science and Technology Program of Yulin City under Grant CXY-2022-91.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Candès, E.J.; Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.* **2009**, *9*, 717–772. [[CrossRef](#)]
2. Cao, F.; Cai, M.; Tan, Y. Image interpolation via low-rank matrix completion and recovery. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1261–1270.
3. Liu, Z.; Vandenberghe, L. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.* **2009**, *31*, 1235–1256. [[CrossRef](#)]
4. Chen, P.; Suter, D. Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1051–1063. [[CrossRef](#)] [[PubMed](#)]
5. Xing, Z.; Zhou, J.; Ge, Z.; Huang, G.; Hu, M. Recovery of high order statistics of PSK signals based on low-rank matrix completion. *IEEE Access* **2023**, *11*, 12973–12986. [[CrossRef](#)]
6. Tsakiris, M.C. Low-rank matrix completion theory via Plücker coordinates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10084–10099. [[CrossRef](#)] [[PubMed](#)]
7. Wang, K.; Chen, Z.; Ying, S.; Xu, X. Low-rank matrix completion via QR-based retraction on manifolds. *Mathematics* **2023**, *11*, 1155. [[CrossRef](#)]
8. Cai, J.-F.; Candès, E.J.; Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982. [[CrossRef](#)]
9. Ma, S.; Goldfarb, D.; Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* **2011**, *128*, 321–353. [[CrossRef](#)]
10. Recht, B.; Fazel, M.; Parrilo, P.A. Guaranteed minimum-rank solution of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **2010**, *52*, 471–501. [[CrossRef](#)]
11. Candès, E.J.; Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theory* **2010**, *56*, 2053–2080. [[CrossRef](#)]
12. Sturm, J.F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **1999**, *11*, 625–653. [[CrossRef](#)]
13. Tütüncü, R.H.; Toh, K.C.; Todd, M.J. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.* **2003**, *95*, 189–217. [[CrossRef](#)]
14. Toh, K.-C.; Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pac. J. Optim.* **2010**, *6*, 615–640.
15. Yang, J.; Yuan, X. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Math. Comput.* **2013**, *82*, 301–329. [[CrossRef](#)]
16. Daubechies, I.; Defrise, M.; Mol, C.D. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.* **2004**, *57*, 1413–1457. [[CrossRef](#)]
17. Chartrand, R. Shrinkage mappings and their induced penalty functions. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014.
18. Peng, D.; Xiu, N.; Yu, J. $S_{1/2}$ regularization methods and fixed point algorithms for affine rank minimization problems. *Comput. Optim. Appl.* **2017**, *67*, 543–569. [[CrossRef](#)]
19. Xu, Z.; Chang, X.; Xu, F.; Zhang, H. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1013–1027. [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.