Stochastic Variance Reduced Gradient for affine rank minimization problem

Ningning Han∗, Juan Nie†, Jian Lu‡, Michael K. Ng§

Abstract

We develop an efficient stochastic variance reduced gradient descent algorithm to solve the affine rank minimization problem consists of finding a matrix of minimum rank from linear measurements. The proposed algorithm as a stochastic gradient descent strategy enjoys a more favorable complexity than full gradients. It also reduces the variance of the stochastic gradient at each iteration and accelerate the rate of convergence. We prove that the proposed algorithm converges linearly in expectation to the solution under a restricted isometry condition. The numerical experiments show that the proposed algorithm has a clearly advantageous balance of efficiency, adaptivity, and accuracy compared with other state-of-the-art greedy algorithms.

Keywords: Low-rank matrix, affine rank minimization, stochastic variance reduced gradient.

1 Introduction

Affine rank minimization problem is a fundamental problem that arises in many practical applications of computer vision, machine learning and signal processing, such as collaborative filtering [1]-[3], image and video processing [4]-[6], phaseless signal recovery [7]-[8], communication system [9]-[11], multi-task learning [12]-[14], etc. Let \( X^* = \{ X^*_{i,j} \} \in \mathbb{R}^{n_1 \times n_2} \) be the ground truth low-rank matrix, and we acquire information about \( X^* \) through a linear mapping \( A : \mathbb{R}^{n_1 \times n_2} \rightarrow \mathbb{R}^m \), i.e., \( y = A(X^*) \) or \( y_\ell = A_\ell(X^*) = \langle A_\ell, X^* \rangle \), \( \ell = 1, \ldots, m \), where \( A_\ell \in \mathbb{R}^{n_1 \times n_2} \) denote the sensing matrices making up the linear mapping \( A(\cdot) \), then the low-rank matrix minimization problem can be formulated as follows:

\[
\min_X \text{rank}(X), \quad \text{subj. to} \quad y = A(X). \tag{1.1}
\]

(1.1) are clearly combinatorial and computationally intractable. Various computationally efficient algorithms for solving (1.1) have been extensively studied. A large majority of algorithms are based on two strategies: convex or non-convex relaxations and greedy iterative algorithms. The renowned advance of relaxations is to replace the optimization problem with the rank function by nuclear norm, namely,

\[
\min_X \|X\|_*, \quad \text{subj. to} \quad y = A(X). \tag{1.2}
\]

For a given \( n \times n \) square matrix, Candés et al prove that if the number \( m \) of sampled entries satisfies \( m \geq Cn^{1.2}r \log n \) for some positive numerical constant \( C \), then with very high probability, most \( n \times n \) matrices of rank \( r \) can be exactly recovered by solving the convex optimization (1.2) [15]. Readers are referred to a series of articles focused on the theoretical analysis [16]-[18] and numerical algorithms [19]-[22] of the nuclear norm approach.

The singular values indicate clear geometric interpretations and should be regularize differently. As the nuclear norm penalizes each singular value equally, the nuclear norm may not be a good surrogate to the

∗School of Mathematical Sciences, Tiangong University, Tianjin, 300387, China email: ningninghan@tiangong.edu.cn.
†Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, 518060, China. email: niejuan0522@163.com.
‡Corresponding author. Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, 518060, China. email: jianlu@szu.edu.cn.
§Department of Mathematics, the University of Hong Kong, Pokfulam, Hong Kong SAR. email: mng@maths.hku.hk.
rank function. To get a more accurate and robust approximation to the rank function, a novel method called truncated nuclear norm regularization \cite{23,25} is proposed, which only minimized the smallest \(p\) singular values to recover the low-rank component. Note that all the existing nonconvex penalty functions are concave and their gradients are decreasing functions, iterative reweighted nuclear norms are proposed to solve low-rank matrix completion \cite{25,26,27}. Inspired by the paradigm of \(\ell_p\) quasi-norm \((0 < p < 1)\) in compressive sensing, some try to expand this concept to the traditional nuclear norm \cite{28,29}, which can approximate the rank function better.

Alternating minimization \cite{31} is also widely used for affine rank minimization problem. Among these algorithms, a symbolic work, known as the factorization \(Z = UV^T\), where \(U \in \mathbb{R}^{n \times r}\) and \(V \in \mathbb{R}^{n \times r}\), explicitly optimize on the manifold of rank \(r\) matrices. The renowned advance of relaxations is to replace the optimization problem \((1.2)\) with the following non-convex problem

\[
\min_{U,V} \frac{1}{2} \|y - A(UV^T)\|_F^2. \tag{1.3}
\]

Two representatives alternating minimization schemes for solving model \((1.3)\) are the power factorization algorithm \cite{32} and the low-rank matrix fitting algorithm \cite{33}. In \cite{34}, the authors propose an alternating steepest descent and a scaled variant scaled alternating steepest descent, where an exact line-search is incorporated to update the solutions of the model \((1.3)\). Yao et al. \cite{35} propose a general nonconvex loss instead of \(\ell_1\) loss to improve robustness of matrix factorization. For a nonconvex function \(f(UV^T)\) w.r.t. \(U\) and \(V\), the bi-factored gradient descent (BFGD) algorithm, as an efficient first-order method is proposed to operate directly on the \(U, V\) factors \cite{36}. Li et al. \cite{37} study the problem of recovering a low-rank matrix from a number of random linear measurements that are corrupted by outliers, where authors propose a nonsmooth nonconvex formulation of the problem and enforce the low-rank property of the solution by using a factored representation of the matrix variable. An simple iterative algorithm based on a Gauss-Newton is proposed to solve low rank matrix recovery, where a key property of Gauss-Newton Matrix Recovery is that it implicitly keeps the factor matrices approximately balanced throughout its iterations \cite{38}. The authors in \cite{39} formulate matrix completion as a feasibility problem and an alternating projection algorithm is devised to find a feasible point in the intersection of the low-rank constraint set and fidelity constraint set. Scaled gradient descent (ScaledGD) viewed as preconditioned or diagonally-scaled gradient descent has also been developed, where the preconditioners are adaptive and iteration-varying with a minimal computational overhead \cite{40}. In addition, Riemannian conjugated gradient method minimizes the least-square distance on the sampling set over the Riemannian manifold of fixed-rank matrices and the algorithm is an adaptation of classical non-linear conjugate gradients \cite{41}.

Greedy algorithms such as iterative hard thresholding (IHT) are another class of popular approaches, one advantage of greedy approaches is that they are considerably low computational complexity. A representative strategy called singular value thresholding, which produces a sequence of matrices, and at each step mainly performs a soft-thresholding operation on the singular values of matrix \cite{42}. Similarly, a fast singular value projection algorithm \cite{43} has been proposed where the hard thresholding operator is employed to penalize singular values. It has been shown in \cite{43} that if the sensing operator \(A(\cdot)\) satisfies constrained restricted isometry property, then iterative hard thresholding with appropriate constant stepsize is guaranteed to recover any low rank matrix. Tanner et al. \cite{44} introduce an efficient alternating projection algorithm, where the proposed algorithm uses an adaptive stepsize calculated to be exact for a restricted subspace. Furthermore, the authors develop a conjugate gradient iterative hard thresholding family of algorithms, which can balance the low per iteration complexity of simple hard thresholding algorithms with the fast asymptotic convergence rate of employing the conjugate gradient method \cite{45}. A family of Riemannian optimization algorithms for low rank matrix has also been introduced for low rank matrix recovery, which are first interpreted as iterative hard thresholding algorithms with subspace projections \cite{46}.

Recent technological advances in data collection and storage raise new challenges in large-scale signal processing problems that essentially involve optimization over particularly large-scale data. Stochastic gradient descent as effective and efficient optimization methods has been widely used for training machine learning models on massive datasets. Stochastic gradient descent (SGD) algorithms have been applied to solve low-rank matrix recovery \cite{47}, where these algorithms avoid computing the full gradient and possess favorable properties in solving large-scale problems especially when computing the full gradient is expensive or prohibitive. Note that stochastic gradient descent iterates with the inherent variance, which preserves slow
convergence asymptotically. To remedy this problem, stochastic variance reduced gradient (SVRG) [50] has been introduced as an explicit variance reduction strategy for stochastic gradient descent. The main aim of this paper is to exploit IHT and SVRG, and propose a new algorithm to solve affine rank minimization problem. The advantage of this algorithm is to reduce the variance of the stochastic gradient at each iteration and accelerate the rate of convergence. We prove the proposed algorithm converges linearly for affine rank minimization problem and conduct a series of numerical experiments to illustrate that the proposed algorithms have a clearly advantageous balance of efficiency, adaptivity and accuracy compared with other state-of-the-art algorithms.

Algorithm 1 SVRG for affine rank minimization problem

**Input:** \( K, n, r, y, A, \epsilon, \eta \)

**Output:** \( \hat{X} = \bar{X}_k \)

**Initialize:** \( \bar{X}_0 \)

for \( k = 0, 1, \ldots, K - 1 \) do

\[ g_k = \frac{1}{m} \sum_{\ell=1}^{m} \nabla f_{\ell}(\bar{X}_k) \]

\[ X_0 = \bar{X}_k \]

for \( t = 0, \ldots, n - 1 \) do

Randomly pick \( i_t \in \{1, \ldots, m\} \)

\[ W_t = X_t - \eta \left( \nabla f_{i_t}(X_t) - \nabla f_{i_t}(\bar{X}_k) + g_k \right) \]

\[ X_{t+1} = \mathcal{H}_r(W_t) \]

end for

\[ \bar{X}_{k+1} = X_n \]

If \( \|y - A(X_{k+1})\|_2^2 \leq \epsilon \) or \( \|\bar{X}_{k+1} - \bar{X}_k\|_F^2 \leq \epsilon \), exit

end for

2 SVRG algorithm for affine rank minimization problem

The cost function \( F(X) \) can be defined by

\[ F(X) = \frac{1}{m} \|y - A(X)\|_2^2 = \frac{1}{m} \sum_{\ell=1}^{m} (y_\ell - \langle A_\ell, X \rangle)^2 = \frac{1}{m} \sum_{\ell=1}^{m} f_\ell(X), \]

we perform the following minimization to recover \( X^* \)

\[ \min_{X} F(X), \text{ subj. to } \text{rank}(X) \leq r. \]  

(2.4)

A standard method for solving (2.4) is gradient descent, which updates the iterations by

\[ X_t = X_{t-1} - \eta \nabla F(X_{t-1}) = X_{t-1} - \frac{\eta}{m} \sum_{\ell=1}^{m} \nabla f_{\ell}(X_{t-1}). \]

Note that gradient descent strategy requires evaluation of \( m \) derivatives, which is computationally expensive. A popular modification is stochastic gradient descent, where we can choose a random training sample set \( i_t \) of size \( |i_t| \) from \( \{1, 2, \ldots, m\} \) and the variable is updated by

\[ X_t = X_{t-1} - \eta \nabla f_{i_t}(X_{t-1}), \]

where \( f_{i_t}(X) = \frac{1}{|i_t|} \sum_{\ell \in i_t} (y_\ell - \langle A_\ell, X \rangle)^2 \). Although the computational cost of stochastic gradient descent is smaller than full gradient descent strategy, it introduces variance due to random selection. In this paper, we employ stochastic variance reduced gradient (SVRG) [50] to reduce the variance and accelerate convergence rate.
The proposed stochastic variance reduced gradient for affine rank minimization problem (SVRG-ARM) is provided in Algorithm 1. The outer loop computes a full gradient $g_k$, which is designed to reduce the variance caused by stochastic gradient descent. The inner loop first selects randomly an index set $i_t$ from the set $\{1, \ldots, m\}$ and then compute the stochastic variance reduced gradient associated with the selected index set. Note that $E(\nabla f_{i_t}(X_t)) = g_k$ in the inner loop, we can force the gradient to be unbiased by letting gradient as $\nabla f_{i_t}(X_t) - (\nabla f_{i_t}(\tilde{X}_k) - g_k)$ and then move the solution along the gradient direction to obtain solution $W_t$. The current solution $W_t$ needs to be projected onto the constraint space $\mathcal{M}_r$ via hard thresholding operator $\mathcal{H}_r(W_t)$, where $\mathcal{M}_r = \{X \in \mathbb{R}^{n_1 \times n_2} : \text{rank}(X) = r\}$. Figure 1 shows a geometric description of SVRG-ARM.

3 Linear Convergence analysis of SVRG-ARM

In this section, we provide linear convergence analysis of the proposed SVRG-ARM algorithm. It should be pointed out that the linear convergence condition is not necessarily optimal at present times, which can be relaxed with perhaps plenty of rooms to improve. We first present the key preliminary results needed in the subsequent analyses.

Definition 3.1. (Restricted isometry property (RIP) [17]). Let $A(\cdot) : \mathbb{R}^{n_1 \times n_2} \rightarrow \mathbb{R}^m$ be a linear map of $n_1 \times n_2$ matrices to vectors of length $m$. For every integer $1 \leq r \leq \min(n_1, n_2)$, the restricted isometry constant $\delta_r$ of $A(\cdot)$ is defined as the smallest number such that

\begin{equation}
(1 - \delta_r)\|X\|_F^2 \leq \frac{1}{m}\|A(X)\|_2^2,
\end{equation}

\begin{equation}
\frac{1}{|i_t|}\|A_{i_t}(X)\|_2^2 \leq (1 + \delta_r)\|X\|_F^2, \quad i_t \in \{1, \ldots, m\},
\end{equation}

holds for all matrices $X$ of rank at most $r$.

Lemma 3.1. For any two low-rank matrices $X$ and $Y$, let $\Gamma$ be a space spanned by $X$ and $Y$ and the rank of any matrix in $\Gamma$ is at most $s$, then

\begin{equation}
\langle X - Y, \nabla F(X) - \nabla F(Y) \rangle \geq 2(1 - \delta_s)\|X - Y\|_F^2,
\end{equation}

and

\begin{equation}
F(Y) \geq F(X) + \langle \nabla F(X), Y - X \rangle + (1 - \delta_s)\|X - Y\|_F^2.
\end{equation}
Proof. It follows from the RIP that

\[
\langle X - Y, \nabla F(X) - \nabla F(Y) \rangle = \frac{2}{m} \left( X - Y, \sum_{\ell=1}^{m} A_\ell \langle A_\ell, X - Y \rangle \right)
\]

\[
= \frac{2}{m} \sum_{\ell=1}^{m} \langle A_\ell, X - Y \rangle^2
\]

\[
= \frac{2}{m} \| A(X - Y) \|_F^2
\]

\[
\geq 2 (1 - \delta_s) \| X - Y \|_F^2.
\]

Together with (3.6), we have

\[
\langle \nabla f_i (X), X - Y \rangle \leq 2 (1 + \delta_s) \| X - Y \|_F^2.
\]

The equivalent conditions (iv) and (iii) in Lemma 2 (30) implies

\[
F(Y) \geq F(X) + \langle \nabla F(X), Y - X \rangle + (1 - \delta_s) \| X - Y \|_F^2.
\]

\[
\square
\]

Lemma 3.2. For any two low-rank matrices X and Y, let \( \Gamma \) be a space spanned by X and Y and the rank of any matrix in \( \Gamma \) is at most \( s \). Then, we have

\[
\| P_\Gamma (\nabla f_i (X) - \nabla f_i (Y)) \|_F^2 \leq 2 (1 + \delta_s) \langle X - Y, \nabla f_i (X) - \nabla f_i (Y) \rangle.
\]

Proof. In view of \( f_i (X) = \frac{1}{|t_\ell|} \sum_{\ell \in t} (y_\ell - \langle A_\ell, X \rangle)^2 \) and \( \nabla f_i (X) = \frac{2}{|t_\ell|} \sum_{\ell \in t} A_\ell (\langle A_\ell, X \rangle - y_\ell) \), we obtain

\[
\frac{1}{|t_\ell|} \| A_\ell (X - Y) \|_F^2 = \frac{1}{|t_\ell|} \sum_{\ell \in t} \langle A_\ell, X - Y \rangle^2
\]

\[
= \frac{1}{|t_\ell|} \left( \sum_{\ell \in t} A_\ell \langle A_\ell, X - Y \rangle, X - Y \right)
\]

\[
= \frac{1}{2} \langle \nabla f_i (X) - \nabla f_i (Y), X - Y \rangle.
\]

Together with with (3.6), we have

\[
\langle \nabla f_i (X) - \nabla f_i (Y), X - Y \rangle \leq 2 (1 + \delta_s) \| X - Y \|_F^2.
\]

Since \( f_i (\cdot) \) is a convex function, the equivalent conditions (3) and (0) in Lemma 4 (30) implies

\[
\| \nabla f_i (Y) - \nabla f_i (X) \|_F \leq 2 (1 + \delta_s) \| X - Y \|_F.
\]

(3.10)

Define the function \( h_i (Z) = f_i (Z) - \langle \nabla f_i (X), Z \rangle \) and follow from (3.10),

\[
\| \nabla h_i (Z_1) - \nabla h_i (Z_2) \|_F = \| \nabla f_i (Z_1) - \nabla f_i (Z_2) \|_F \leq 2 (1 + \delta_s) \| Z_1 - Z_2 \|_F
\]

holds for \( \forall Z_1, Z_2 \in \Gamma \). Using the equivalent conditions (0) and (2) in Lemma 4 (30) about the convex function \( h_i (\cdot) \), we have

\[
h_i (Z_1) - h_i (Z_2) - \langle \nabla h_i (Z_2), Z_1 - Z_2 \rangle \leq (1 + \delta_s) \| Z_1 - Z_2 \|_F^2.
\]

(3.11)

For \( \forall Z \in \Gamma \), according to the definitions of \( f_i (Z), f_i (X), \) and \( \nabla f_i (X) \), we obtain

\[
h_i (Z) - h_i (X) = f_i (Z) - f_i (X) - \langle \nabla f_i (X), Z - X \rangle
\]

\[
= \frac{1}{|t_\ell|} \sum_{\ell \in t} \langle A_\ell, Z - X \rangle^2 \geq 0.
\]

(3.12)
Define $Z = Y - \frac{1}{2(1 + \delta_s)} P_T \nabla h_i(t)(Y)$. Applying (3.12) and (3.11) gives

$$h_i(X) \leq h_i(Z) = h_i \left( Y - \frac{1}{2(1 + \delta_s)} P_T \nabla h_i(t)(Y) \right) \leq h_i(Y) + \left\langle \nabla h_i(Y), -\frac{1}{2(1 + \delta_s)} P_T \nabla h_i(t)(Y) \right\rangle + \frac{1}{4(1 + \delta_s)} \| P_T \nabla h_i(t)(Y) \|^2_F$$

$$= h_i(Y) - \frac{1}{4(1 + \delta_s)} \| P_T \nabla h_i(t)(Y) \|^2_F$$

Based on the definition of $h_i(\cdot)$ and the above inequality, we have

$$\frac{1}{4(1 + \delta_s)} \| P_T \nabla h_i(t)(Y) \|^2_F = \frac{1}{4(1 + \delta_s)} \| P_T \nabla f_i(t)(Y) - \nabla f_i(t)(X) \|^2_F \leq h_i(Y) - h_i(X) = f_i(Y) - f_i(X) - \langle \nabla f_i(X), Y - X \rangle$$

Similarly, interchanging the role of $Y$ and $X$ leads to

$$\frac{1}{4(1 + \delta_s)} \| P_T (\nabla f_i(t)(X) - \nabla f_i(t)(Y)) \|^2_F \leq f_i(X) - f_i(Y) - \langle \nabla f_i(Y), X - Y \rangle$$

Taking the summation, we derive

$$\| P_T (\nabla f_i(t)(X) - \nabla f_i(t)(Y)) \|^2_F \leq 2 \left( 1 + \delta_s \right) \langle X - Y, \nabla f_i(X) - \nabla f_i(Y) \rangle$$

Lemma 3.3. For any two low-rank matrices $X$ and $Y$, let $\Gamma$ be a space spanned by $X$ and $Y$ and the rank of any matrix in $\Gamma$ is at most $s$ and $\eta \leq \frac{1}{1 + \delta_s}$. Then, we have

$$\| X - Y - \eta \nabla F(X) - \nabla F(Y) \|_F \leq \sqrt{1 - 2 \left( 1 + \delta_s \right) (2\eta - 2\eta^2 (1 + \delta_s))} \| X - Y \|_F$$

Proof.

$$\| X - Y - \eta \nabla F(X) - \nabla F(Y) \|_F^2 = \| X - Y \|^2_F + \eta^2 \| \nabla F(X) - \nabla F(Y) \|^2_F - 2\eta \langle X - Y, \nabla F(X) - \nabla F(Y) \rangle$$

$$\leq \| X - Y \|^2_F + \eta^2 (1 + \delta_s) \langle X - Y, \nabla F(X) - \nabla F(Y) \rangle - 2\eta \langle X - Y, \nabla F(X) - \nabla F(Y) \rangle$$

$$= \| X - Y \|^2_F - (2\eta - 2\eta^2 (1 + \delta_s)) \langle X - Y, \nabla F(X) - \nabla F(Y) \rangle$$

where the first inequality follows from Lemma 3.3 with taking $i = \{1, \ldots, m\}$ and the last inequality follows from Lemma 3.1.

Lemma 3.4. For any two low-rank matrices $X$ and $Y$, let $\Gamma$ be a space spanned by $X$ and $Y$ and the rank of any matrix in $\Gamma$ is at most $s$. Denote $i_t$ be the index randomly selected from $\{1, \ldots, m\}$ and $\eta \leq \frac{1}{1 + \delta_s}$, then we have

$$\mathbb{E}_{i_t} \| X - Y - \eta \nabla f_i(t)(X) - \nabla f_i(t)(Y) \|_F \leq \sqrt{1 - 2 \left( 1 + \delta_s \right) (2\eta - 2\eta^2 (1 + \delta_s))} \| X - Y \|_F$$

Proof.

$$\mathbb{E}_{i_t} \| X - Y - \eta \nabla f_i(t)(X) - \nabla f_i(t)(Y) \|_F^2 = \| X - Y \|^2_F + \eta^2 \mathbb{E}_{i_t} \| \nabla f_i(t)(X) - \nabla f_i(t)(Y) \|^2_F - 2\eta \mathbb{E}_{i_t} \langle X - Y, \nabla f_i(t)(X) - \nabla f_i(t)(Y) \rangle$$

$$\leq \| X - Y \|^2_F + 2\eta^2 (1 + \delta_s) \mathbb{E}_{i_t} \langle X - Y, \nabla f_i(t)(X) - \nabla f_i(t)(Y) \rangle - 2\eta \mathbb{E}_{i_t} \langle X - Y, \nabla f_i(t)(X) - \nabla f_i(t)(Y) \rangle$$

$$= \| X - Y \|^2_F - (2\eta - 2\eta^2 (1 + \delta_s)) \mathbb{E}_{i_t} \langle X - Y, \nabla f_i(t)(X) - \nabla f_i(t)(Y) \rangle$$

$$= \| X - Y \|^2_F - (2\eta - 2\eta^2 (1 + \delta_s)) \langle X - Y, \nabla F(X) - \nabla F(Y) \rangle$$

$$\leq \| X - Y \|^2_F - 2 \left( 1 + \delta_s \right) (2\eta - 2\eta^2 (1 + \delta_s)) \| X - Y \|^2_F$$

6
where the first inequality is based on Lemma 3.2, the last equality follows from the fact that $\nabla f_i (X)$ is an unbiased estimation to $\nabla F$, i.e., $E[\nabla f_i (x_i) | x_i] = \nabla F(x_i)$ and the last inequality follows from Lemma 3.1. The desired result follows by applying Jensen inequality $(\mathbb{E}Z)^2 \leq \mathbb{E}(Z)^2$.

**Theorem 3.1.** Assume that $X^*$ is the optimal solution to (2.4), the linear mapping $A$ satisfies RIP defined in Definition 3.1 with $\delta_{3r} \leq \frac{1}{4}$, and the step size satisfies

$$
\frac{6 - 6\delta_{3r} - \sqrt{71\delta_{3r}^2 - 72\delta_{3r} + 1}}{12 - 12\delta_{3r}^2} < \eta < \frac{6 - 6\delta_{3r} + \sqrt{71\delta_{3r}^2 - 72\delta_{3r} + 1}}{12 - 12\delta_{3r}^2}
$$

then SVRG-ARM converges linearly in expectation:

$$
E_{i_t} \| \bar{X}_{k} - X^* \|_F \leq k^r_{3r} \| \bar{X}_0 - X^* \|_F
$$

where $\rho_{3r} = 2\sqrt{1 - 2(1 - \delta_{3r})(2\eta - 2\eta^2 (1 + \delta_{3r}))}$ and $\kappa_{3r} = \frac{-3\rho_{3r}^{\kappa_{3r}} + 3\rho_{3r}^2 + 2\rho_{3r}}{1 - \rho_{3r}} < 1$.

**Proof.** Note that Eckart–Young theorem guarantees that $X_{t+1}$ is the rank $r$ matrix nearest to $W_t$ in the Frobenius norm, we have $\|X_{t+1} - W_t\|_F^2 \leq \|X^* - W_t\|_F^2$. It follows that

$$
\|X_{t+1} - X^*\|_F^2 = \|X_{t+1} - X^* + X^* - W_t\|_F^2 - \|X^* - W_t\|_F^2 - 2\langle X_{t+1} - X^*, X^* - W_t \rangle
$$

$$
\leq \|X_{t+1} - X^*, W_t - X^*\|
$$

$$
\leq 2\left( X_{t+1} - X^*, X_t - X^* - \eta \left( \nabla f_i (X_t) - \nabla f_i (\tilde{X}_k) \right) + g_k \right) - X^*ight)
$$

$$
= 2 \left( X_{t+1} - X^*, X_t - X^* - \eta \left( \nabla f_i (X_t) - \nabla f_i (X^*) \right) \right)
$$

$$
- 2 \left( X_{t+1} - X^*, \tilde{X}_k - X^* - \eta \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) \right)
$$

$$
+ 2 \left( X_{t+1} - X^*, \tilde{X}_k - X^* - \eta \left( \nabla F (\tilde{X}_k) - \nabla F (X^*) \right) \right)
$$

where the fourth equality follows from $g_k = \nabla F (\tilde{X}_k)$ and $\nabla F (X^*) = 0$. Denote $\Omega_t$ as the subspace spanned by $X_{t+1}$, $X_t$ and $X^*$ and $\Omega'_t$ as the subspace spanned by $X_{t+1}$, $\tilde{X}_k$ and $X^*$. Define $P_{\Omega_t} : \mathbb{R}^{n_1 \times n_2} \rightarrow \Omega_t$ as the orthogonal projection onto $\Omega_t$. Obviously, $P_{\Omega_t}(X_{t+1}) = X_{t+1}$, $P_{\Omega_t}(X_t) = X_t$, $P_{\Omega_t}(X^*) = X^*$, $P_{\Omega'_t}(X_{t+1}) = X_{t+1}$, $P_{\Omega'_t}(\tilde{X}_k) = \tilde{X}_k$ and $P_{\Omega'_t}(X^*) = X^*$. The rank of any matrix in $\Omega_t$ and $\Omega'_t$ is at most $3r$. Consequently, we obtain

$$
\|X_{t+1} - X^*\|_F^2 \leq 2 \left( X_{t+1} - X^*, X_t - X^* - \eta P_{\Omega_t} (\nabla f_i (X_t) - \nabla f_i (X^*)) \right)
$$

$$
- 2 \left( X_{t+1} - X^*, \tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla f_i (\tilde{X}_k) - \nabla f_i (X^*)) \right)
$$

$$
+ 2 \left( X_{t+1} - X^*, \tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla F (\tilde{X}_k) - \nabla F (X^*)) \right)
$$

$$
\leq 2 \|X_{t+1} - X^*\|_F (\|X_t - X^* - \eta P_{\Omega_t} (\nabla f_i (X_t) - \nabla f_i (X^*))\|_F + \|\tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla f_i (\tilde{X}_k) - \nabla f_i (X^*))\|_F + \|\tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla F (\tilde{X}_k) - \nabla F (X^*))\|_F).
$$

Canceling $\|X_{t+1} - X^*\|_F$ in the above inequality gives the inequality

$$
\|X_{t+1} - X^*\|_F \leq 2 (\|X_t - X^* - \eta P_{\Omega_t} (\nabla f_i (X_t) - \nabla f_i (X^*))\|_F
$$

$$
+ \|\tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla f_i (\tilde{X}_k) - \nabla f_i (X^*))\|_F + \|\tilde{X}_k - X^* - \eta P_{\Omega'_t} (\nabla F (\tilde{X}_k) - \nabla F (X^*))\|_F).
$$

(3.13)
As defined $\eta < \frac{6-6\delta_3r\sqrt{1+\delta_3r}+72\delta_3r+1}{12-12\delta_3r}$, note that $i_t$ determines the solution $X_{t+1}$, taking the expectation on both sides of (3.13) yields,

$$
\mathbb{E}_{i_t} \| X_{t+1} - X^* \|_F \leq 2 \mathbb{E}_{i_t} \| X_t - X^* - \eta P_{3r} (\nabla f_{i_t}(X_t) - \nabla f_{i_t}(X^*)) \|_F
$$

$$+
\mathbb{E}_{i_t} \| \tilde{X}_k - X^* - \eta P_{3r} \left( \nabla f_{i_t} \left( \tilde{X}_k \right) - \nabla f_{i_t}(X^*) \right) \|_F
$$

$$+
\| \tilde{X}_k - X^* - \eta P_{3r} \left( \nabla F \left( \tilde{X}_k \right) - \nabla F(X^*) \right) \|_F
$$

$$\leq 2(\sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))) \| X_t - X^* \|_F
$$

$$+ \sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))) \| \tilde{X}_k - X^* \|_F
$$

$$+ \sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))) \| \tilde{X}_k - X^* \|_F
$$

$$= 2 \sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))) \| X_t - X^* \|_F
$$

$$+ 4 \sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))) \| \tilde{X}_k - X^* \|_F$$

where the second inequality follows from Lemma 3.4 and Lemma 3.3. By recursively applying the above inequality over $t$, and noting that $X_k = X_0$ and $X_{k+1} = X_n$, we can obtain

$$
\mathbb{E}_{i_t} \| \tilde{X}_{k+1} - X^* \|_F = \mathbb{E}_{i_t} \| X_n - X^* \|_F \leq (\rho_3^{n+1} + 2\rho_3^n + \cdots + 2\rho_3) \| \tilde{X}_k - X^* \|_F
$$

$$\leq \frac{-3\rho_3^{n+1} + \rho_3^n + 2\rho_3}{1 - \rho_3} \| \tilde{X}_k - X^* \|_F,$$

where $\rho_3 = 2\sqrt{1 - 2(1 - \delta_3r)} (2\eta - 2\eta^2 (1 + \delta_3r))$. Since $\delta_3r < \frac{1}{11}$ and $\frac{6-6\delta_3r\sqrt{1+\delta_3r}+72\delta_3r+1}{12-12\delta_3r} < \eta < \frac{6-6\delta_3r\sqrt{1+\delta_3r}+72\delta_3r+1}{12-12\delta_3r}$, we have $\frac{-3\rho_3^{n+1} + \rho_3^n + 2\rho_3}{1 - \rho_3} < 1$. The linear convergence of SVRG-ARM algorithm for affine rank minimization problem follows immediately. \qed

### 4 Complexity analysis

This section contains the result about complexity analysis of SVRG-ARM. We first present the key lemmas needed in the subsequent analyses of the number of iterations for obtaining accuracy of $\epsilon$.

**Lemma 4.1.** Let $\Gamma$ be a space spanned by $X$ and $X^*$, and the rank of any matrix in $\Gamma$ is at most $s$. Then we have

$$
\mathbb{E}[\| P_\Gamma (\nabla f_{i_t}(X) - \nabla f_{i_t}(X^*)) \|_F^2] \leq 4 (1 + \delta_s) (F(X) - F(X^*)).
$$

**Proof.** For any $i_t \in \{1, 2, \ldots, m\}$, and $X \in \mathbb{R}^{n_1 \times n_2}$, we define

$$
\varphi_{i_t}(X) = f_{i_t}(X) - f_{i_t}(X^*) - \langle \nabla f_{i_t}(X^*), X - X^* \rangle
$$

(4.14)

Then, we can get a similar inequality as in (3.14)

$$
\| \nabla \varphi_{i_t}(X) - \nabla \varphi_{i_t}(Y) \|_F = \| \nabla f_{i_t}(X) - \nabla f_{i_t}(Y) \|_F \leq 2(1 + \delta_s) \| X - Y \|_F^2.
$$

(4.15)

Since $\nabla \varphi_{i_t}(X^*) = 0$, we have $\varphi_{i_t}(X^*) = \min_X \varphi_{i_t}(X)$. Together with (4.15), and the equivalent conditions (0) and (2) in Lemma 4.2 [30], it results in

$$
0 = \varphi_{i_t}(X^*) \leq \varphi_{i_t} \left( X - \frac{\Pi_\Gamma \left( 1 \over 2(1 + \delta_s) \nabla \varphi_{i_t}(X) \right) \right)
$$

$$\leq \varphi_{i_t} \left( X - \left( \nabla \varphi_{i_t}(X), \Pi_\Gamma \left( 1 \over 2(1 + \delta_s) \nabla \varphi_{i_t}(X) \right) \right) \right) + (1 + \delta_s) \| \Pi_\Gamma (\nabla \varphi_{i_t}(X)) \|_F^2
$$

$$= \varphi_{i_t} \left( X - \frac{1}{4(1 + \delta_s)} \| \Pi_\Gamma (\nabla \varphi_{i_t}(X)) \|_F^2 \right).
$$

8
From the definition (4.14), we have
\[ \| \mathcal{P}_i ( \nabla f_i (X) - \nabla f_i (X^*) ) \|_F^2 \leq 4 (1 + \delta_\tau) (f_i (X) - f_i (X^*) - \langle \nabla f_i (X^*), X - X^* \rangle). \]

Taking expectation with respect to \( i \), we get
\[
\mathbb{E} \| \mathcal{P}_i ( \nabla f_i (X) - \nabla f_i (X^*) ) \|_F^2 \leq 4 (1 + \delta_\tau) (F(X) - F(X^*) - \langle \nabla F (X^*), X - X^* \rangle)
= 4 (1 + \delta_\tau) (F(X) - F(X^*)) ,
\]
where the last equality follows from the fact \( \nabla F (X^*) = 0 \).

\[ \mathbf{Lemma 4.2.} \] Let \( X^* \) be the optimal solution to (2.4). Given a low-rank matrix \( X \), where \( \text{rank}(X) = r \), let \( \Lambda \) be a space spanned by \( X \) and \( X^* \), and the rank of any matrix in \( \Lambda \) is at most \( \tau \), then we have
\[ \| \mathcal{P}_\Lambda (\nabla F (X)) \|_F^2 \geq 2 \frac{(1 - \delta_\tau)}{1 + \delta_\tau} (F(X) - F(X^*)) \]

\[ \text{Proof.} \] We first note that
\[ \langle X - X^*, \nabla F (X) - \nabla F (X^*) \rangle = \langle X - X^*, \mathcal{P}_\Lambda (\nabla F (X) - \nabla F (X^*)) \rangle \leq \| X - X^* \|_F \| \mathcal{P}_\Lambda (\nabla F (X)) \|_F \]

Together with (3.7), we have
\[ \| \mathcal{P}_\Lambda (\nabla F (X)) \|_F^2 \geq 4 (1 - \delta_\tau)^2 \| X - X^* \|_F^2 \quad (4.16) \]

Let \( i_t \in \{1, \ldots, m\} \) in (3.9), we have
\[ \langle \nabla F (X) - \nabla F (X^*), X - X^* \rangle \leq 2 (1 + \delta_\tau) \| X - X^* \|_F^2. \]

The equivalent conditions (3) and (2) in Lemma 4 (30) implies
\[ F(X) \leq F(X^*) + \langle \nabla F(X^*), X - X^* \rangle + (1 + \delta_\tau) \| X - X^* \|_F^2 = F(X^*) + (1 + \delta_\tau) \| X - X^* \|_F^2. \quad (4.17) \]

Combining (4.16) and (4.17) yields the desired result
\[ \| \mathcal{P}_\Lambda (\nabla F (X)) \|_F^2 \geq 4 \frac{(1 - \delta_\tau)^2}{1 + \delta_\tau} (F(X) - F(X^*)) \]

\[ \mathbf{Lemma 4.3.} \] Denote \( \Omega_t \) as the subspace spanned by \( \tilde{X}_k \), \( X_t \) and \( X^* \), and the rank of any matrix in \( \Omega_t \) is at most \( 3r \). Let \( V_t = \nabla f_i (X_t) - \nabla f_i (\tilde{X}_k) + \nabla F (\tilde{X}_k) \) as defined in Algorithm 1, then we have
\[ \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} (V_t) \|_F^2 \leq 8 (1 + \delta_{3r}) (F(X_t) - F(X^*)) + \frac{32 \delta_{3r}}{1 + \delta_{3r}} \left( F (\tilde{X}_k) - F (X^*) \right). \]

\[ \text{Proof.} \] By the definition of \( \mathcal{P}_{\Omega_t} (V_t) \), we have
\[
\mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} (V_t) \|_F^2 = \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \nabla f_i (X_t) - \nabla f_i (\tilde{X}_k) + \nabla F (\tilde{X}_k) \right) \|_F^2
\]
\[
= \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (X_t) - \nabla f_i (X^*) \right) - \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) + \nabla F (\tilde{X}_k) \right) \|_F^2
\]
\[
\leq 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (X_t) - \nabla f_i (X^*) \right) \right) \|_F^2 + 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) - \nabla F (\tilde{X}_k) \right) \|_F^2
\]
\[
= 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (X_t) - \nabla f_i (X^*) \right) \right) \|_F^2 + 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) \right) \|_F^2
\]
\[
= 2 \mathbb{E}_{i_t} \mathcal{P}_{\Omega_t} \left( \nabla F (\tilde{X}_k) \right) \|_F^2 - 4 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) \right) \|_F^2
\]
\[
= 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \nabla f_i (X_t) - \nabla f_i (X^*) \right) \|_F^2 + 2 \mathbb{E}_{i_t} \| \mathcal{P}_{\Omega_t} \left( \left( \nabla f_i (\tilde{X}_k) - \nabla f_i (X^*) \right) \right) \|_F^2
\]
\[
\leq 8 (1 + \delta_{3r}) (F(X_t) - F(X^*)) + 8 (1 + \delta_{3r}) \left( F (\tilde{X}_k) - F (X^*) \right) - \frac{8 \delta_{3r}^2}{1 + \delta_{3r}} \left( F (\tilde{X}_k) - F (X^*) \right)
= 8 (1 + \delta_{3r}) (F(X_t) - F(X^*)) + \frac{32 \delta_{3r}}{1 + \delta_{3r}} \left( F (\tilde{X}_k) - F (X^*) \right). \]
Theorem 4.1. Assume that $X^*$ is the optimal solution to (2.4), the linear mapping $A$ satisfies RIP defined in Definition 3.1 with $\delta_{3r} \leq \frac{1}{29}$, and the step size satisfies

$$\frac{2(1 + \delta_{3r}) \sqrt{1 - \delta_{3r}} - \sqrt{68\delta_{3r} + 388\delta_{3r}^2 - 60\delta_{3r} + 4}}{(16\delta_{3r}^2 + 96\delta_{3r} + 16) \sqrt{1 - \delta_{3r}}} \leq \eta \leq \frac{2(1 + \delta_{3r}) \sqrt{1 - \delta_{3r}} + \sqrt{68\delta_{3r} + 388\delta_{3r}^2 - 60\delta_{3r} + 4}}{(16\delta_{3r}^2 + 96\delta_{3r} + 16) \sqrt{1 - \delta_{3r}}}$$

then the sequence produced by SVRG-ARM satisfies

$$\mathbb{E} \left( F(\tilde{X}_{k+1}) - F(X^*) \right) \leq \beta_{3r} \mathbb{E} \left( F(\tilde{X}_k) - F(X^*) \right)$$

where $\beta_{3r} = \left( \mu_{3r}^n + \frac{\nu_{3r}(1 - \mu_{3r}^n)}{1 - \mu_{3r}} \right) < 1$, $\mu_{3r} = \frac{1 + \delta_{3r}}{1 - \delta_{3r}} - 2\eta (1 + \delta_{3r}) (1 - 4\eta (1 + \delta_{3r}))$ and $\nu_{3r} = 32\delta_{3r}\eta^2$.

Proof. Let $\Omega_t$ be a space spanned by $\tilde{X}_k$, $X_t$ and $X^*$. Since $\text{rank}(\tilde{X}_k) = r$, $\text{rank}(X_t) = r$ and $\text{rank}(X^*) = r$, the rank of any matrix in $\Omega_t$ is at most $3r$. Then

$$\mathbb{E}_t \|X_{t+1} - X^*\|^2_F = \mathbb{E}_t \|X_t - \eta V_t - X^*\|^2_F = \mathbb{E}_t \|X_t - \eta P_{\Omega_t}(V_t) - X^*\|^2_F$$

where the inequality follows from Lemma 4.3 and (3.8). Taking $i_t = \{1, \ldots, m\}$ in (3.9), the equivalent conditions (3) and (2) in Lemma 4 (3.9), we can derive

$$\|X_{t+1} - X^*\|^2_F \geq (F(X_{t+1}) - F(X^*)) / (1 + \delta_{3r})$$

Combining (4.18) and (4.19), we obtain

$$\mathbb{E}_t (F(X_{t+1}) - F(X^*)) \leq \left( \frac{1 + \delta_{3r}}{1 - \delta_{3r}} - 2\eta (1 + \delta_{3r}) (1 - 4\eta (1 + \delta_{3r})) \right) (F(X_t) - F(X^*)) + 32\delta_{3r}\eta^2 \left( F(\tilde{X}_k) - F(X^*) \right)$$

By recursively applying the above inequality over $t$, and noting that $\tilde{X}_k = X_0$ and $\tilde{X}_{k+1} = X_n$, we can obtain

$$\mathbb{E} \left( F(\tilde{X}_{k+1}) - F(X^*) \right) \leq \left( \mu_{3r}^n + \frac{\nu_{3r}(1 - \mu_{3r}^n)}{1 - \mu_{3r}} \right) \mathbb{E} \left( F(\tilde{X}_k) - F(X^*) \right)$$

where $\mu_{3r} = \frac{1 + \delta_{3r}}{1 - \delta_{3r}} - 2\eta (1 + \delta_{3r}) (1 - 4\eta (1 + \delta_{3r}))$ and $\nu_{3r} = 32\delta_{3r}\eta^2$.

By choosing $\delta_{3r} \leq \frac{1}{20}$ and

$$\frac{2(1 + \delta_{3r}) \sqrt{1 - \delta_{3r}} - \sqrt{68\delta_{3r} + 388\delta_{3r}^2 - 60\delta_{3r} + 4}}{(16\delta_{3r}^2 + 96\delta_{3r} + 16) \sqrt{1 - \delta_{3r}}} \leq \eta \leq \frac{2(1 + \delta_{3r}) \sqrt{1 - \delta_{3r}} + \sqrt{68\delta_{3r} + 388\delta_{3r}^2 - 60\delta_{3r} + 4}}{(16\delta_{3r}^2 + 96\delta_{3r} + 16) \sqrt{1 - \delta_{3r}}}$$

we have $\beta_{3r} = \left( \mu_{3r}^n + \frac{\nu_{3r}(1 - \mu_{3r}^n)}{1 - \mu_{3r}} \right) < 1$. $\square$
By Theorem 4.1, we have \( \mathbb{E} \left( F \left( \tilde{X}_k \right) - F (X^*) \right) \leq \beta_3^k \mathbb{E} \left( F \left( \tilde{X}_0 \right) - F (X^*) \right) \). To obtain accuracy of \( \epsilon \), i.e., \( \mathbb{E} \left( F \left( \tilde{X}_k \right) - F (X^*) \right) \leq \epsilon \), SVRG-ARM needs to take \( k = \mathcal{O} \left( \log \left( 1/\epsilon \right) \right) \) outer loops. The computational complexity of the proposed algorithm mainly includes two parts: the computation of gradients and singular value decompositions. The complexity of calculating gradients is \( \mathcal{O} \left( m + nb \right) \), where \( n \) is the number of inner loops and \( b = \max \{ i_0, \ldots, i_{n-1} \} \). Besides, a singular value decomposition is required in each iteration to project the variable \( W^t \) back onto the rank \( r \) matrix feasible solution space and the corresponding complexity can be \( \mathcal{O} \left( r^3 \right) \), where \( r \) is the rank of low-rank matrix [46]. Therefore, the overall computational complexity
of SVRG-ARM is $O\left((m + nb + r^3)\log\left(\frac{1}{\varepsilon}\right)\right)$. If $f_i(x)$ is $L$-Lipschitz smooth and $F(x)$ is $\mu$-strongly convex, deterministic full gradient descent method needs $O\left(\sqrt{\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$ iterations to find an $\varepsilon$-accurate solution, where $\kappa$ is the condition number $L/\mu$ \cite{53, 54}. The overall computational complexity of deterministic full gradient descent method is $O\left((m + r^3)\sqrt{\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$. Thus SVRG-ARM presents a significant improvement over deterministic full gradient descent method when $\kappa$ is large, which has also been validated by numerical experiments in Section 5.1.

![Comparison of matrix completion algorithms for image inpainting.](image)

Figure 4: Comparison of matrix completion algorithms for image inpainting. (a) Original image. (b) Observed image with missing pixels. (c)-(h) Recovered images by SVT, SVP, NIHT, CGIHT, StoIHT, SVRG-ARM.

5 Numerical experiments

In this section, we present numerical results on synthetic and real data to validate the proposed algorithm. For comprehensive and complete comparisons, we first compare performance within the class of gradient descent algorithms including singular value thresholding (SVT) \cite{42}, singular value projection (SVP) \cite{43}, normalized iterative hard thresholding (NIHT) \cite{44}, conjugate gradient iterative hard thresholding (CGIHT) \cite{46}, stochastic iterative hard thresholding (StoIHT) \cite{47}. Among these algorithms, SVP is the simplest iterative hard thresholding gradient descent algorithm with fixed stepsize, while SVT is the iterative soft-thresholding gradient descent algorithm. NIHT is the modified iterative hard thresholding gradient descent algorithm with an adaptive stepsize. SVP and NIHT need to calculate the full gradient at each iteration. CGIHT generates the current estimate along the Riemannian conjugate gradient descent. StoIHT is based on stochastic gradient descent, and SVRG-ARM is designed to reduce the variance of stochastic gradient descent. In addition, we utilize Barzilai-Borwein (BB) \cite{51, 52} method to automatically calculate step sizes, where we set the step size $\eta_k = \|\tilde{X}_k - \tilde{X}_{k-1}\|_F^2 / (n(\tilde{X}_k - \tilde{X}_{k-1}, g_k - g_{k-1}))$ at each iteration.

Then the overall performance of SVRG-ARM in terms of execution-time and frequency of exact recovery is compared with other state-of-the-art algorithms including matrix factorization based method solved by ScaledASD \cite{33}, nuclear norm minimization (NNM) based method solved by augmented Lagrange multiplier method \cite{22}, iterative reweighted nuclear norm (IRNN) \cite{25, 26, 27}, truncated nuclear norm regularization.
(TNNR) [23] [25], $\ell_p$ quasi-norm ($0 < p < 1$) [28] [29].

The associated matlab codes can be downloaded from the authors’ webpages or provided by authors in personal communication. A matlab implementation of the proposed algorithm is also available at https://www.dropbox.com/s/9gte2as7gcar180/SVRG-ARM.zip?dl=0.

![Figure 5: Comparison of matrix completion algorithms for image inpainting. (a) Original image. (b) Observed image with missing pixels. (c)-(h) Recovered images by SVT, SVP, NIHT, CGIHT, StoIHT, SVRG-ARM.](image)

5.1 Performance comparison within the class of gradient descent algorithms

In this subsection, we conduct comparisons about matrix completion. The matrix completion as a classical affine rank minimization problem aims to recover a low-rank matrix from partially observed entries. We generate $n_1 \times n_2$ matrices of rank $r$ as a product of a $n_1 \times r$ matrix and a $r \times n_2$ matrix, whose entries follow the Gaussian distributions. The locations of observed indices are sampled uniformly at random. Let $\rho$ be the sample ratio of observed entries over $n_1 \times n_2$. 

The first performance metric refers to the frequency rate of exact recovery. An exact recovery is recorded whenever \( \| \hat{X} - X \|_F / \| X \|_F \leq 10^{-3} \), where \( \hat{X} \) denotes the estimate of original low-rank matrix \( X \). We fix the matrix size to be \( n_1 = n_2 = 50 \), set the sample ratio \( \rho \) to be 0.5 and vary rank \( r \) to investigate the probability of recovery success. Each algorithm is tested for 100 (random) trials for every rank \( r \). Figure 6(a) shows the frequency of exact recovery as a function of the rank. First, the recovery ability can be reflected by critical sparsity. The critical sparsity is the maximal sparsity level of the desired signal at which the exact recovery is ensured. Indeed, higher critical sparsity represents better empirical recovery performance. Figure 6(a) reveals that the critical sparsity of SVRG-ARM is larger than that of other methods. The second metric is the convergence speed. In this experiment, the parametric setting is \( n_1 = n_2 = 50 \), \( \rho = 0.5 \), \( r = 4 \). As shown in Figure 6(b), except for SVT, the recovery accuracies of all other algorithms are almost identical. The convergence of SVRG-ARM is faster than other methods to reach the same optimality. These experiments suggest that SVRG-ARM outperforms StoIHT in both frequency of exact recovery and running time. It demonstrates the theoretical findings about variance reduced gradient, namely the conclusion that SVRG can reduce the variance introduced by stochastic gradient descent and accelerate the rate of convergence. To test the robustness to noise, we add the Gaussian noise with zero mean and standard deviation varying from 0 to 0.4 to low-rank matrix. Relative errors of all algorithms versus noise level are shown in Figure 6(c). As shown, SVP, NIHT, CGIHT, StoIHT and SVRG-ARM are in the same level and SVRG-ARM slightly outperforms other methods.

Figure 6: (a) Frequency of exact recovery as a function of rank. (b) Convergence speed. (c) Normalized mean square error as a function of noise level

To further validate the effectiveness of SVRG-ARM, we check the recovery ability as a function of rank \( r \) and proportion of sample ratio \( \rho \). We fix the matrix size to be \( n_1 = n_2 = 50 \) and vary rank \( r \) and sample ratio \( \rho \) to investigate the probability of recovery success. For each pair \((r, \rho)\), we simulate 100 test instances. Figure 7 shows the fraction of perfect recovery for each pair (black = 0 and white =1). As known, the smaller the percentage of missing values and the smaller the rank, the larger the region of correct recovery is. It is clear that the performance of our method SVRG-ARM is better than that of other methods.

We then present color image completion results. The size of the first image pepper is 512 × 512, the set of observed entries are generated randomly and the percentage of observed entries is 0.5. The comparison is to apply matrix completion method to the luminance channel. Both the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are provided for the comparison. We find PSNR and SSIM by the proposed SVRG-ARM algorithm is better than that of other methods. As shown in Figure 4, in the rectangle region, it can be seen that SVRG-ARM generates high-level visual quality with sharper edges and richer textures in comparison with other methods. The size of the second image facade is 517 × 493, the set of observed entries are generated randomly and the percentage of observed entries is 0.4. The quantitative comparisons show that SVRG-ARM can provide larger PSNR and SSIM values than those by other methods. In addition, Figure 5 shows that the reconstructed image by SVRG-ARM has higher quality edges with proper sharpness and limited artifacts. The experimental results verify that SVRG-ARM outperforms other gradient descent algorithms in terms of both synthetic and real data.
5.2 Overall comparison with state-of-the-art algorithms

Presented here are comparisons among SVRG-ARM and state-of-the-art techniques such as NNM, $\ell_p$ quasi-norm, ScaledASD, IRNN, and TNNR in terms of frequency of exact recovery, convergence speed and robustness. In the first experiment, rank varies from 3 to 15 with matrix size of $n_1 = n_2 = 50$ and sample ratio $\rho = 0.5$. As shown in Figure 6(a), IRNN and TNNR present better recovery performances than SVRG-ARM. In the following experiments, we set $n_1 = n_2 = 50$, $\rho = 0.5$ and $r = 8$. For comparison, the particular rank selection $r = 5$ is used for NNM. For the execution-time comparison, ScaledASD achieves the best convergence speed, and however SVRG-ARM presents a better recovery accuracy than ScaledASD. To test the robustness to noise, we add the Gaussian noise with zero mean and standard deviation varying from 0 to 0.4 to low-rank matrix. Relative errors of all algorithms versus noise level are shown in Figure 6(c). As shown, SVRG-ARM is more robust than other algorithms. Experimental results suggest that no algorithm is consistently superior for all cases. But SVRG-ARM is observed to have obviously advantageous balance of efficiency, accuracy and robustness compared with other algorithms.

We then compare phase transitions of low-rank matrix completion using different methods, where the recovery ability as a function of rank $r$ and proportion of sample ratio $\rho$ is investigated. Successful recovery is indicated by white and failure by black. Results are averaged over 100 independent trials. Figure 7 shows that SVRG-ARM still delivers reasonable performance better than that of NNM, $\ell_p$ quasi-norm, ScaledASD, though slightly underperforms that of IRNN and TNNR.

Finally, we conduct image completions to compare different methods. The size of the first image flower is $512 \times 480$, the set of observed entries are generated randomly and the percentage of observed entries is 0.5. Two common image quality evaluation criteria PSNR and SSIM are still employed to reflect the image recovery quality. Figure 8 shows that our algorithm achieves the highest PSNR and SSIM among all methods. From the rectangle region, it can be also observed that SVRG-ARM provides high-level visual quality with sharper edges and richer textures. To further illustrate the effectiveness of the proposed method, we show

![Figure 7: Phase transition of low-rank matrix completion using (a) NNM. (b) $\ell_p$ quasi-norm. (c) ScaledASD. (d) IRNN. (e) TNNR. (f) SVRG-ARM.](image-url)
the reconstructed results of image baboon by different methods in Figure 9. In this experiment, the size of the image is 512 × 512, the set of observed entries are generated randomly and the percentage of observed entries is 0.4. This experiment manifests that SVRG-ARM can obtain good results especially referring to edges (high frequency details). Numerical results about image completions demonstrate the effectiveness of SVRG-ARM among different low-rank matrix completion algorithms.

![Figure 8: Comparison of matrix completion algorithms for image completion. (a) Original image. (b) Observed image with missing pixels. (c)-(h) Recovered images by NNM, ℓ_p quasi-norm, ScaledASD, IRNN, TNNR, SVRG-ARM.](image)

6 Conclusion

We introduce a particularly simple yet highly efficient stochastic variance reduced gradient descent algorithm to solve the affine rank minimization problem consists of finding a matrix of minimum rank from linear measurements. We prove that the proposed algorithm converges linearly in expectation to the solution
under a restricted isometry condition. It should be pointed out that the linear convergence condition is not necessarily optimal at present times, which can be relaxed with perhaps plenty of rooms to improve. The proposed algorithm is observed to have obviously advantageous balance of efficiency, adaptivity, and accuracy compared with other state-of-the-art greedy algorithms. A matlab implementation of the proposed algorithm is also available at https://www.dropbox.com/s/9gte2as7gcarl80/SVRG-ARM.zip?dl=0.

![Figure 9: Comparison of matrix completion algorithms for image inpainting.](image_url)

- (a) Original image
- (b) Observed image with missing pixels
- (c) NNM 21.5856/0.6405
- (d) $\ell_p$ quasi-norm 22.4057/0.6831
- (e) ScaledASD 23.4571/0.7465
- (f) IRNN 23.7856/0.7602
- (g) TNNR 23.8627/0.7638
- (h) SVRG-ARM 24.2060/0.7796

Figure 9: Comparison of matrix completion algorithms for image inpainting. (a) Original image. (b) Observed image with missing pixels. (c)-(h) Recovered images by NNM, $\ell_p$ quasi-norm, ScaledASD, IRNN, TNNR, SVRG-ARM.

Acknowledgments

This work is supported in part by GuangDong Basic and Applied Basic Research Foundation under grant 2021A1515110530, the Foundation for Distinguished Young Talents of Guangdong under grant 2021KQNCX075,
National Natural Science Foundation of China under grants U21A20455, 61972265, 11871348 and 61373087, the Natural Science Foundation of Guangdong Province of China under grant 2020B1515310008, the Educational Commission of Guangdong Province of China undergrant 2019KZDZX1007, and the Guangdong Key Laboratory of Intelligent Information Processing, China. M. Ng’s research is supported in part by the HKRGC GRF 12300218, 12300519, 17201020, 17300021, C1013-21GF, C7004-21GF, and Joint NSFC-RGC N-HKU76921.

References

[1] J. D. Rennie and N. Srebro, Fast maximum margin matrix factorization for collaborative prediction, in Proceedings of the 22nd international conference on Machine learning, 2005, 713–719.

[2] G. Takács, I. Pilászy, B. Németh, and D. Tikk, Investigation of various matrix factorization methods for large recommender systems, 2008 IEEE International Conference on Data Mining Workshops, 2008: 553-562.

[3] N. Rao, H.-F. Yu, Ravikumar, P. Ravikumar, and I. S. Dhillon, Collaborative filtering with graph information: Consistency and scalable methods. Advances in neural information processing systems, 2015, 28.

[4] H. Ji, C. Q. Liu, Z. W. Shen, and Y. H. Xu, Robust video denoising using Low rank matrix completion, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010: 1791-1798.

[5] R. S. Cabral, J. P. Costeira, and A. Bernardino, Matrix Completion for Multi-label Image Classification, Advances in neural information processing systems, 2011, 24.

[6] P. J. Shin, P. E. Larson, M. A. Ohliger, M. Elad, J. M. Pauly, D. B. Vigneron, M. Lustig, Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion, Magnetic resonance in medicine, 2014, 72 (4): 959-970.

[7] E. J. Candès, T. Strohmer, V. Voroninski, Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming, Communications on Pure and Applied Mathematics, 2013, 66 (8): 1241-1274.

[8] E. J. Candès, Y. C. Eldar, T. Strohmer, V. Voroninski, Phase retrieval via matrix completion, SIAM review, 2015, 57 (2): 225-251.

[9] B. Li, A. P. Petropulu, W. Trappe, Optimum co-design for spectrum sharing between matrix completion based MIMO radars and a MIMO communication system[J]. IEEE Transactions on Signal Processing, 2016, 64 (17): 4562-4575.

[10] D. S. Kalogerias, A. P. Petropulu, Matrix completion in colocated MIMO radar: Recoverability, bounds & theoretical guarantees, IEEE Transactions on Signal Processing, 2013, 62 (2): 309-321.

[11] Z. Qin, Y. Liu , Y. Gao, M. Elkashlan, A. Nallanathan, Wireless powered cognitive radio networks with compressive sensing and matrix completion, IEEE Transactions on Communications, 2016, 65 (4): 1464-1476.

[12] T. K. Pong, P. Tseng, S. Ji, J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning, SIAM Journal on Optimization, 2010, 20 (6): 3465-3489.

[13] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, Advances in neural information processing systems, 2006, 19.

[14] R. Zhang, H. Zhang, X. Li, Robust multi-task learning with flexible manifold constraint, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(6): 2150-2157.

[15] E. J. Candès, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics, 2009, 9 (6): 717-772.
[16] E. J. Candés, T. Tao, The power of convex relaxation: Near-optimal matrix completion, IEEE Transactions on Information Theory, 2010, 56 (5): 2053-2080.

[17] B. Recht, F. Maryam, P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Review, 2010, 52 (3): 471-501.

[18] B. Recht, A simpler approach to matrix completion, Journal of Machine Learning Research, 2011, 12 (12): 3413–3430.

[19] J. F. Cai, E. J. Candés, Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization, 2010, 20 (4): 1956-1982.

[20] Z. Liu, L. Vandenberghe, Interior-point method for nuclear norm approximation with application to system identification, SIAM Journal on Matrix Analysis and Applications, 2010 31 (3): 1235-1256.

[21] S. Q. Ma, D. Goldfarb, L. F. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, Mathematical Programming, 2011, 128 (1): 321-353.

[22] Z. C. Lin, M. M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv preprint arXiv:1009.5055, 2010.

[23] Y. Hu, D. B. Zhang, J. P. Ye, X. L. Li, X. F. He, Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35 (9): 2117-2130.

[24] D. B. Zhang, Y. Hu, J. P. Ye, X. L. Li, X. F. He, Matrix completion by truncated nuclear norm regularization, IEEE Conference on Computer Vision and Pattern Recognition, 2012, 2192-2199.

[25] C. Y. Lu, J. H. Tang, S. C. Yan, Z. C. Lin, Generalized nonconvex nonsmooth low-rank minimization, IEEE Conference on Computer Vision and Pattern Recognition, 2014, 4130-4137.

[26] K. Mohan, M. Fazel, Iterative reweighted algorithms for matrix rank minimization, The Journal of Machine Learning Research, 2012, 13 (1): 3441-3473.

[27] M. Fornasier, H. Raurut, R. Ward, Low-rank matrix recovery via iteratively reweighted least squares minimization, SIAM Journal on Optimization, 2011, 21 (4): 1614-1640.

[28] F. P. Nie, H. Huang, C. Ding, Low-rank matrix recovery via efficient schatten p-norm minimization, AAAI conference on artificial intelligence, 2012.

[29] Y. Xie, S. H. Gu, Y. Liu, W. M. Zuo, W. S. Zhang, L .Zhang, Weighted Schatten p-norm minimization for image denoising and background subtraction, IEEE Transactions on Image Processing, 2016, 25 (10): 4842-4857.

[30] X. Y, Zhou, On the fenchel duality between strong convexity and lipschitz continuous gradient, arXiv preprint arXiv 1803.06573, 2018.

[31] R. Escalante, R. Marcos, Alternating projection methods, Society for Industrial and Applied Mathematics, 2011.

[32] J.P. Haldar, D. Hernando, Rank-constrained solutions to linear matrix equations using PowerFactorization, IEEE Signal Processing Letters, 2009, 16 (7): 584–587.

[33] W. Z. Wen, W. T. Yin, Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, Mathematical Programming Computation , 2012, 4 (4): 333-361.

[34] J. Tanner, K. Wei, Low rank matrix completion by alternating steepest descent methods, Applied and Computational Harmonic Analysis, 2016, 40 (2) : 417-429.
[35] Q. M. Yao, K. James, Scalable robust matrix factorization with nonconvex loss, Advances in Neural Information Processing Systems 2018, 31.

[36] D. Y. Park, A. Kyrillidis, C. Caramanis, S. Sanghavi, Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably, SIAM Journal on Imaging Sciences, 2018, 11 (4): 2165-2204.

[37] X. Li, Z. H. Zhu, A. M. So, R. Vidal, Nonconvex robust low-rank matrix recovery, SIAM Journal on Optimization, 2020, 30 (1): 660-686.

[38] P. Zilber, B. Nadler, GNMR: A provable one-line algorithm for low rank matrix recovery, SIAM Journal on Mathematics of Data Science, 2022 4 (2): 909-934.

[39] X. Jiang, Z. M. Zhong, X. Z. Liu, H. C. So, Robust matrix completion via alternating projection, IEEE Signal Processing Letters, 2017 24 (5): 579-583.

[40] T. Tong, C. Ma, Y. J. Chi, Accelerating Ill-Conditioned Low-Rank Matrix Estimation via Scaled Gradient Descent, Journal of Machine Learning Research, 2021 22: 150-1.

[41] B. Vandereycken, Low-rank matrix completion by Riemannian optimization, SIAM Journal on Optimization, 2013 23 (2): 1214-1236.

[42] J. F. Cai, E. J. Candès, Z. W Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on optimization, 2010 20 (4): 1956-1982.

[43] P. Jain, R. Meka, I. Dhillon, Guaranteed rank minimization via singular value projection, Advances in Neural Information Processing Systems, 2010 23.

[44] J. Tanner, K Wei, Normalized iterative hard thresholding for matrix completion, SIAM Journal on Scientific Computing, 2013 35 (5): S104-S125.

[45] J. D. Blanchard, J. Tanner, K. Wei, CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion, Information and Inference: A Journal of the IMA, 2015 4 (5): 289-327.

[46] K. Wei, J. F. Cai, T. F. Chan, S. Y. Leung, Guarantees of Riemannian optimization for low rank matrix recovery, SIAM Journal on Matrix Analysis and Applications, 2016 37 (3): 1198-1222.

[47] N. Nguyen, D. Needell, T. Woolf, Linear convergence of stochastic iterative greedy algorithms with sparse constraints, IEEE Transactions on Information Theory, 2017, 63 (11): 6869-6895.

[48] A. Nitanda, Stochastic proximal gradient descent with acceleration techniques, Advances in Neural Information Processing Systems, 2014, 27.

[49] H. Lin, J. Mairal, Z. Harchaoui, A universal catalyst for first order optimization, Advances in Neural Information Processing Systems, 2015, 8.

[50] R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, Advances in neural information processing systems, 2013 (26).

[51] J. Barzilai, J. Borwein, Two-point step size gradient methods, IMA journal of numerical analysis, 1988 8 (1): 141-148.

[52] C. Tan, S. Q. Ma, Y. H. Dai, Y. Q. Qian, Barzilai-borwein step size for stochastic gradient descent, Advances in neural information processing systems, 2016 (29).

[53] A. Nitanda, Stochastic proximal gradient descent with acceleration techniques, Advances in neural information processing systems, 2014: 1574–1582.

[54] H. Lin, J. Mairal, Z. Harchaoui, A universal catalyst for first-order optimization, Advances in neural information processing systems, 2015: 3366–3374.