A Unified Variance Reduction-Based Framework for Nonconvex Low-Rank Matrix Recovery

Lingxiao Wang [*1] **Xiao Zhang** [*1] **Quanquan Gu** [1]

Abstract

We propose a generic framework based on a new stochastic variance-reduced gradient descent algorithm for accelerating nonconvex low-rank matrix recovery. Starting from an appropriate initial estimator, our proposed algorithm performs projected gradient descent based on a novel semi-stochastic gradient specifically designed for low-rank matrix recovery. Based upon the mild restricted strong convexity and smoothness conditions, we derive a projected notion of the restricted Lipschitz continuous gradient property, and prove that our algorithm enjoys linear convergence rate to the unknown low-rank matrix with an improved computational complexity. Moreover, our algorithm can be employed to both noiseless and noisy observations, where the (near) optimal sample complexity and statistical rate can be attained respectively. We further illustrate the superiority of our generic framework through several specific examples, both theoretically and experimentally.

1. Introduction

Low-rank matrix recovery problem has been extensively studied during the past decades, due to its wide range of applications, such as collaborative filtering (Srebro et al., 2004; Rennie & Srebro, 2005) and multi-label learning (Cabral et al., 2011; Xu et al., 2013). The objective of low-rank matrix recovery is to estimate the unknown low-rank matrix $\mathbf{X}^* \in \mathbb{R}^{d_1 \times d_2}$ from partial observations, such as a set of linear measurements in matrix sensing or a subset of its entries in matrix completion. Significant efforts have been made to estimate low-rank matrices, among which one of the most prevalent approaches is nuclear norm relaxation based optimization (Srebro et al., 2004; Candès & Tao, 2010; Rohde et al., 2011; Recht et al., 2010; Negahban & Wainwright, 2011; 2012; Gui & Gu, 2015). While such convex relaxation based methods enjoy a rigorous theoretical guarantee to recover the unknown low-rank matrix, due to the nuclear norm regularization/minimization, these algorithms involve a singular value decomposition at each iteration, whose time complexity is $O(d^3)$ to recover a $d \times d$ matrix. Hence, they are computationally very expensive.

In order to address the aforementioned computational issue, recent studies (Keshavan et al., 2009; 2010; Jain et al., 2013a; Jain & Netrapalli, 2014; Hardt, 2014; Hardt & Wootters, 2014; Hardt et al., 2014; Zhao et al., 2015; Chen & Wainwright, 2015; Sun & Luo, 2015; Zheng & Lafferty, 2015; 2016; Tu et al., 2015; Bhojanapalli et al., 2015; Park et al., 2016b; Wang et al., 2016) have been carried out to perform factorization on the matrix space, which naturally ensures the low-rankness of the produced estimator. Although this matrix factorization technique converts the previous optimization problem into a nonconvex one, which is more difficult to analyze, it significantly improves the computational efficiency.

However, for large-scale matrix recovery, such nonconvex optimization approaches are still computationally expensive, because they are based on gradient descent or alternating minimization, which involve the time-consuming calculation of full gradient at each iteration. De Sa et al. (2014) developed a stochastic gradient descent approach for Gaussian ensembles, but the sample complexity (i.e., number of measurements or observations required for exact recovery) of their algorithm is not optimal. Recently, Jin et al. (2016) and Zhang et al. (2017b) proposed stochastic gradient descent algorithms for noiseless matrix completion and matrix sensing, respectively. Although these algorithms achieve linear rate of convergence and improved computational complexity over aforementioned deterministic optimization based approaches, they are limited to specific low-rank matrix recovery problems, and unable to be extended to more general problems and settings.

In this paper, inspired by the idea of variance reduction for stochastic gradient (Schmidt et al., 2013; Konečnỳ & Richtárik, 2013; Johnson & Zhang, 2013; Defazio et al.,

[*]Equal contribution [1]Department of Computer Science, University of Virginia, Charlottesville, Virginia, USA. Correspondence to: Quanquan Gu <qg5w@virginia.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

2014a;b; Mairal, 2014; Xiao & Zhang, 2014; Konečnỳ et al., 2014; Reddi et al., 2016; Allen-Zhu & Hazan, 2016; Chen & Gu, 2016; Zhang & Gu, 2016), we propose a unified stochastic gradient descent framework with variance reduction for low-rank matrix recovery, which integrates both optimization-theoretic and statistical analyses. To the best of our knowledge, this is the first unified accelerated stochastic gradient descent framework for low-rank matrix recovery with strong convergence guarantees. With a desired initial estimator given by a general initialization algorithm, we show that our algorithm achieves linear convergence rate and better computational complexity against the state-of-the-art algorithms. The contributions of our work are further highlighted as follows:

1. We develop a generic stochastic variance-reduced gradient descent algorithm for low-rank matrix recovery, which can be applied to various low rank-matrix estimation problems, including matrix sensing, noisy matrix completion and one-bit matrix completion. In particular, for noisy matrix sensing, it is guaranteed to linearly converge to the unknown low-rank matrix up to the minimax statistical precision (Negahban & Wainwright, 2011; Wang et al., 2016); while for noiseless matrix sensing, our algorithm achieves the optimal sample complexity (Recht et al., 2010; Tu et al., 2015; Wang et al., 2016), and attains a linear rate of convergence. Besides, for noisy matrix completion, it achieves the best-known sample complexity required by nonconvex matrix factorization (Zheng & Lafferty, 2016).

2. At the core of our algorithm, we construct a novel semi-stochastic gradient term, which is substantially different from the one if following the original stochastic variance-reduced gradient using chain rule (Johnson & Zhang, 2013). This uniquely constructed semi-stochastic gradient has not appeared in the literature, and is essential for deriving the minimax optimal statistical rate.

3. Our unified framework is built upon the mild restricted strong convexity and smoothness conditions (Negahban et al., 2009; Negahban & Wainwright, 2011) regarding the objective function. Based on the above mentioned conditions, we derive an innovative projected notion of the restricted Lipschitz continuous gradient property, which we believe is of independent interest for other nonconvex problems to prove sharp statistical rates. We further establish the linear convergence rate of our generic algorithm. Besides, for each specific examples, we verify that the conditions required in the generic setting are satisfied with high probability, which demonstrates the applicability of our framework.

4. Our algorithm has a lower computational complexity compared with existing approaches (Jain et al., 2013a; Zhao et al., 2015; Chen & Wainwright, 2015; Zheng & Lafferty, 2015; 2016; Tu et al., 2015; Bhojanapalli et al., 2015; Park et al., 2016b; Wang et al., 2016). More specifically, to achieve ϵ precision, the gradient complexity[1] of our algorithm is $O\big((N + \kappa^2 b)\log(1/\epsilon)\big)$. Here N denotes the total number of observations, d denotes the dimensionality of the unknown low-rank matrix $\mathbf{X}^*$, b denotes the batch size, and κ denotes the condition number of $\mathbf{X}^*$ (see Section 2 for a detailed definition). In particular, if the condition number satisfies $\kappa \leq N/b$, our algorithm is computationally more efficient than the state-of-the-art generic algorithm in Wang et al. (2016).

Notation. We use $[d]$ and $\mathbf{I}_d$ to denote $\{1, 2, \ldots, d\}$ and $d \times d$ identity matrix respectively. We write $\mathbf{A}^\top \mathbf{A} = \mathbf{I}_{d_2}$, if $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$ is orthonormal. For any matrix $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$, we use $\mathbf{A}_{i,*}$ and $\mathbf{A}_{*,j}$ to denote the i-th row and j-th column of $\mathbf{A}$, respectively. In addition, we use A_{ij} to denote the (i, j)-th element of $\mathbf{A}$. Denote the row space and column space of $\mathbf{A}$ by $\mathrm{row}(\mathbf{A})$ and $\mathrm{col}(\mathbf{A})$ respectively. Let $d = \max\{d_1, d_2\}$, and $\sigma_\ell(\mathbf{A})$ be the ℓ-th largest singular value of $\mathbf{A}$. For vector $\mathbf{x} \in \mathbb{R}^d$, we use $\|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}$ to denote its ℓ_q vector norm for $0 < q < \infty$. Denote the spectral and Frobenius norm of $\mathbf{A}$ by $\|\mathbf{A}\|_2$ and $\|\mathbf{A}\|_F$ respectively. We use $\|\mathbf{A}\|_{\infty,\infty} = \max_{i,j} |A_{ij}|$ to denote the element-wise infinity norm of $\mathbf{A}$, and we use $\|\mathbf{A}\|_{2,\infty}$ to represent the largest ℓ_2-norm of its rows. Given two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if there exists a constant $0 < C_1 < \infty$ such that $a_n \leq C_1 b_n$. Note that other notations are defined throughout the paper.

2. Methodology

In this section, we present our generic stochastic gradient descent algorithm with variance reduction as well as several illustrative examples.

2.1. Stochastic Variance-Reduced Gradient for Low-Rank Matrix Recovery

First, we briefly introduce the general problem setup for low-rank matrix recovery. Suppose $\mathbf{X}^* \in \mathbb{R}^{d_1 \times d_2}$ is an unknown rank-r matrix. Let the singular value decomposition (SVD) of $\mathbf{X}^*$ be $\mathbf{X}^* = \overline{\mathbf{U}}^* \mathbf{\Sigma}^* \overline{\mathbf{V}}^{*\top}$, where $\overline{\mathbf{U}}^* \in \mathbb{R}^{d_1 \times r}$, $\overline{\mathbf{V}}^* \in \mathbb{R}^{d_2 \times r}$ are orthonormal matrices, and $\mathbf{\Sigma}^* \in \mathbb{R}^{r \times r}$ is a diagonal matrix. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ be the sorted nonzero singular values of $\mathbf{X}^*$, and denote the condition number of $\mathbf{X}^*$ by κ, i.e., $\kappa = \sigma_1/\sigma_r$. Besides, let $\mathbf{U}^* = \overline{\mathbf{U}}^* (\mathbf{\Sigma}^*)^{1/2}$ and $\mathbf{V}^* = \overline{\mathbf{V}}^* (\mathbf{\Sigma}^*)^{1/2}$. Recall that we aim to recover $\mathbf{X}^*$ through a collection of N observations or measurements. Let $\mathcal{L}_N : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ be the sample loss function, which evaluates the fitness of any matrix $\mathbf{X}$ associated with the total N observations. Then the low-rank

[1] Gradient complexity is defined as the number of gradients calculated in total.

matrix recovery problem can be formulated as follows:

$$\min_{\mathbf{X}\in\mathbb{R}^{d_1\times d_2}} \mathcal{L}_N(\mathbf{X}) := \frac{1}{N}\sum_{i=1}^N \ell_i(\mathbf{X}), \quad \text{subject to} \quad \mathbf{X}\in\mathcal{C},\ \text{rank}(\mathbf{X})\le r, \tag{2.1}$$

where $\ell_i(\mathbf{X})$ measures the fitness of $\mathbf{X}$ associated with the i-th observation. Here $\mathcal{C}\subseteq\mathbb{R}^{d_1\times d_2}$ is a feasible set, such that $\mathbf{X}^*\in\mathcal{C}$. In order to more efficiently estimate the unknown low-rank matrix, following Jain et al. (2013a); Tu et al. (2015); Zheng & Lafferty (2016); Park et al. (2016a); Wang et al. (2016), we decompose $\mathbf{X}$ as $\mathbf{U}\mathbf{V}^\top$ and consider the following nonconvex optimization problem via matrix factorization:

$$\min_{\mathbf{U}\in\mathcal{C}_1,\mathbf{V}\in\mathcal{C}_2} \mathcal{L}_N(\mathbf{U}\mathbf{V}^\top) := \frac{1}{N}\sum_{i=1}^N \ell_i(\mathbf{U}\mathbf{V}^\top), \tag{2.2}$$

where $\mathcal{C}_1\subseteq\mathbb{R}^{d_1\times r}, \mathcal{C}_2\subseteq\mathbb{R}^{d_2\times r}$ are the rotation-invariant sets induced by $\mathcal{C}$. Recall $\mathbf{X}^*$ can be factorized as $\mathbf{X}^* = \mathbf{U}^*\mathbf{V}^{*\top}$, then we need to make sure that $\mathbf{U}^*\in\mathcal{C}_1$ and $\mathbf{V}^*\in\mathcal{C}_2$. Besides, it can be seen from (2.2) that the optimal solution is not unique in terms of rotation. In order to deal with such identifiability issue, following Tu et al. (2015); Zheng & Lafferty (2016); Park et al. (2016b), we consider the following regularized optimization problem:

$$\min_{\mathbf{U}\in\mathcal{C}_1,\mathbf{V}\in\mathcal{C}_2} F_N(\mathbf{U},\mathbf{V}) := \mathcal{L}_N(\mathbf{U}\mathbf{V}^\top) + \mathcal{R}(\mathbf{U},\mathbf{V}),$$

where the regularization term is defined as $\mathcal{R}(\mathbf{U},\mathbf{V}) = \|\mathbf{U}^\top\mathbf{U}-\mathbf{V}^\top\mathbf{V}\|_F^2/8$. We further decompose the objective function $F_N(\mathbf{U},\mathbf{V})$ into n components to apply stochastic variance-reduced gradient descent:

$$F_N(\mathbf{U},\mathbf{V}) := \frac{1}{n}\sum_{i=1}^n F_i(\mathbf{U},\mathbf{V}), \tag{2.3}$$

where we assume $N = nb$, and b denotes batch size, i.e., the number of observations associated with each F_i. More specifically, we have

$$F_i(\mathbf{U},\mathbf{V}) = \mathcal{L}_i(\mathbf{U}\mathbf{V}^\top) + \mathcal{R}(\mathbf{U},\mathbf{V}),$$
$$\mathcal{L}_i(\mathbf{U}\mathbf{V}^\top) = \frac{1}{b}\sum_{j=1}^b \ell_{ij}(\mathbf{U}\mathbf{V}^\top). \tag{2.4}$$

Therefore, based on (2.3) and (2.4), we are able to apply the stochastic variance-reduced gradient, which is displayed as Algorithm 1. As will be seen in later theoretical analysis, the variance of the proposed stochastic gradient indeed decreases as the iteration number increases, which leads to a faster convergence rate. Let $\mathcal{P}_{\mathcal{C}_i}$ be the projection operator onto the feasible set $\mathcal{C}_i$ in Algorithm 1, where $i\in\{1,2\}$.

Note that our proposed Algorithm 1 is different from the standard stochastic variance-reduced gradient algorithm (Johnson & Zhang, 2013) in several aspects. First, instead of conducting gradient descent directly on $\mathbf{X}$, our algorithm performs alternating stochastic gradient descent on the factorized matrices U and V, which leads to a better computational complexity but a more challenging analysis. Second, we construct a novel semi-stochastic gradient term for $\mathbf{U}$ (resp. $\mathbf{V}$) as $\nabla_{\mathbf{U}}F_{i_t}(\mathbf{U},\mathbf{V}) - \nabla\mathcal{L}_{i_t}(\widetilde{\mathbf{X}})\mathbf{V} + \nabla\mathcal{L}_N(\widetilde{\mathbf{X}})\mathbf{V}$, which is different from $\nabla_{\mathbf{U}}F_{i_t}(\mathbf{U},\mathbf{V}) - \nabla_{\mathbf{U}}F_{i_t}(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}) + \nabla_{\mathbf{U}}F_N(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}})$ if following the original stochastic variance reduced gradient descent (Johnson & Zhang, 2013). This uniquely devised semi-stochastic gradient is essential for deriving the minimax optimal statistical rate. Last but not least, we introduce a projection step to ensure that the estimator produced at each iteration belongs to a feasible set, which is necessary for various low-rank matrix recovery problems. We also note that Reddi et al. (2016); Allen-Zhu & Hazan (2016) recently developed SVRG algorithms for general nonconvex finite-sum optimization problem. However, their algorithms only guarantee a sublinear rate of convergence to a stationary point, and cannot exploit the special structure of low-rank matrix factorization. In stark contrast, our algorithm is able to leverage the structure of the problem and guaranteed to linearly converge to the unknown low-rank matrix instead of a stationary point.

Algorithm 1 Low-Rank Stochastic Variance-Reduced Gradient Descent (LRSVRG)

Input: loss function $\mathcal{L}_N$; step size η; number of iterations S, m; initial solution $(\widetilde{\mathbf{U}}^0, \widetilde{\mathbf{V}}^0)$.
  for: $s = 1, 2, \ldots, S$ **do**
    $\widetilde{\mathbf{U}} = \widetilde{\mathbf{U}}^{s-1}$, $\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}}^{s-1}$, $\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{V}}^\top$
    $\mathbf{U}^0 = \widetilde{\mathbf{U}}$, $\mathbf{V}^0 = \widetilde{\mathbf{V}}$
    for: $t = 0, 1, 2, \ldots, m-1$ **do**
      Randomly pick $i_t \in \{1, 2, \ldots, n\}$
      $\mathbf{U}^{t+1} = \mathcal{P}_{\mathcal{C}_1}\big(\mathbf{U}^t - \eta(\nabla_{\mathbf{U}}F_{i_t}(\mathbf{U}^t,\mathbf{V}^t)$
       $-\nabla\mathcal{L}_{i_t}(\widetilde{\mathbf{X}})\mathbf{V}^t + \nabla\mathcal{L}_N(\widetilde{\mathbf{X}})\mathbf{V}^t)\big)$
      $\mathbf{V}^{t+1} = \mathcal{P}_{\mathcal{C}_2}\big(\mathbf{V}^t - \eta(\nabla_{\mathbf{V}}F_{i_t}(\mathbf{U}^t,\mathbf{V}^t)$
       $-\nabla\mathcal{L}_{i_t}(\widetilde{\mathbf{X}})^\top\mathbf{U}^t + \nabla\mathcal{L}_N(\widetilde{\mathbf{X}})^\top\mathbf{U}^t)\big)$
    end for
    $(\widetilde{\mathbf{U}}^s, \widetilde{\mathbf{V}}^s) = (\mathbf{U}^t, \mathbf{V}^t)$, random $t \in \{0, \ldots, m-1\}$
  end for
Output: $(\widetilde{\mathbf{U}}^S, \widetilde{\mathbf{V}}^S)$.

Algorithm 2 Initialization

Input: loss function $\mathcal{L}_N$; step size τ; iteration number T.
  initialize: $\mathbf{X}_0 = \mathbf{0}$
  for: $t = 1, 2, 3, \ldots, T$ **do**
    $\mathbf{X}_t = \mathcal{P}_r\big(\mathbf{X}_{t-1} - \tau\nabla\mathcal{L}_N(\mathbf{X}_{t-1})\big)$
  end for
  $[\overline{\mathbf{U}}^0, \mathbf{\Sigma}^0, \overline{\mathbf{V}}^0] = \text{SVD}_r(\mathbf{X}_T)$
  $\widetilde{\mathbf{U}}^0 = \overline{\mathbf{U}}^0(\mathbf{\Sigma}^0)^{1/2}$, $\widetilde{\mathbf{V}}^0 = \overline{\mathbf{V}}^0(\mathbf{\Sigma}^0)^{1/2}$
Output: $(\widetilde{\mathbf{U}}^0, \widetilde{\mathbf{V}}^0)$

As will be seen in later analysis, Algorithm 1 requires a good initial solution to guarantee the linear convergence rate. To obtain such an initial solution, we employ the initialization algorithm in Algorithm 2, which is originally proposed in Wang et al. (2016). For any rank-r matrix $\mathbf{X}\in\mathbb{R}^{d_1\times d_2}$, we use $\text{SVD}_r(\mathbf{X})$ to denote its singular

value decomposition. If $\text{SVD}_r(\mathbf{X}) = [\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}]$, we use $\mathcal{P}_r(\mathbf{X}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ to denote the best rank-r approximation of $\mathbf{X}$, or in other words, $\mathcal{P}_r$ denotes the projection operator such that $\mathcal{P}_r(\mathbf{X}) = \text{argmin}_{\text{rank}(\mathbf{Y})\leq r} \|\mathbf{X} - \mathbf{Y}\|_F$.

2.2. Applications to Specific Models

In this subsection, we introduce two examples, which include matrix sensing and matrix completion, to illustrate the applicability of our proposed algorithm (Algorithm 1). The application of our algorithm to one-bit matrix completion can be found in Appendix A. To apply the proposed method, we only need to specify the form of $F_N(\mathbf{U}, \mathbf{V})$ for each specific model, as defined in (2.3).

2.2.1. Matrix Sensing

In matrix sensing (Recht et al., 2010; Negahban & Wainwright, 2011), we intend to recover the unknown matrix $\mathbf{X}^* \in \mathbb{R}^{d_1\times d_2}$ with rank-r from a set of noisy linear measurements such that $\boldsymbol{y} = \mathcal{A}(\mathbf{X}^*) + \boldsymbol{\epsilon}$, where the linear measurement operator $\mathcal{A} : \mathbb{R}^{d_1\times d_2} \to \mathbb{R}^N$ is defined as $\mathcal{A}(\mathbf{X}) = (\langle \mathbf{A}_1, \mathbf{X}\rangle, \langle \mathbf{A}_2, \mathbf{X}\rangle, \ldots, \langle \mathbf{A}_N, \mathbf{X}\rangle)^\top$, for any $\mathbf{X} \in \mathbb{R}^{d_1\times d_2}$. Here N denotes the number of observations, and $\boldsymbol{\epsilon}$ represents a sub-Gaussian noise vector with i.i.d. elements and parameter ν. In addition, for each sensing matrix $\mathbf{A}_i \in \mathbb{R}^{d_1\times d_2}$, it has i.i.d. standard Gaussian entries. Therefore, we formulate $F_N(\mathbf{U}, \mathbf{V})$ for matrix sensing as follows $F_N(\mathbf{U}, \mathbf{V}) = n^{-1}\sum_{i=1}^n F_{\mathcal{S}_i}(\mathbf{U}, \mathbf{V})$, where for each component function, we have $F_{\mathcal{S}_i}(\mathbf{U}, \mathbf{V}) = \|\mathbf{y}_{\mathcal{S}_i} - \mathcal{A}_{\mathcal{S}_i}(\mathbf{U}\mathbf{V}^\top)\|_2^2/(2b) + \mathcal{R}(\mathbf{U}, \mathbf{V})$. Note that $\mathcal{R}(\mathbf{U}, \mathbf{V})$ denotes the regularizer, which is defined in Section 2.1. In addition, $\{\mathcal{S}_i\}_{i=1}^n$ denote the mutually disjoint subsets such that $\cup_{i=1}^n \mathcal{S}_i = [N]$, and $\mathcal{A}_{\mathcal{S}_i}$ is defined as a linear measurement operator $\mathcal{A}_{\mathcal{S}_i} : \mathbb{R}^{d_1\times d_2} \to \mathbb{R}^b$, satisfying $\mathcal{A}_{\mathcal{S}_i}(\mathbf{X}) = (\langle \mathbf{A}_{i_1}, \mathbf{X}\rangle, \langle \mathbf{A}_{i_2}, \mathbf{X}\rangle, \ldots, \langle \mathbf{A}_{i_b}, \mathbf{X}\rangle)^\top$, with corresponding observations $\mathbf{y}_{\mathcal{S}_i} = (y_{i_1}, y_{i_2}, \ldots, y_{i_b})^\top$.

2.2.2. Matrix Completion

For matrix completion with noisy observations (Rohde et al., 2011; Koltchinskii et al., 2011; Negahban & Wainwright, 2012), our primary goal is to recover the unknown low-rank matrix $\mathbf{X}^* \in \mathbb{R}^{d_1\times d_2}$ from a set of randomly observed noisy elements. For example, one commonly-used model is the uniform observation model, which is defined as follows:

$$Y_{jk} := \begin{cases} X^*_{jk} + Z_{jk}, & \text{with probability } p, \\ *, & \text{otherwise,} \end{cases}$$

where $\mathbf{Z} \in \mathbb{R}^{d_1\times d_2}$ is a noise matrix such that each element Z_{jk} follows i.i.d. Gaussian distribution with variance $\nu^2/(d_1 d_2)$, and we call $\mathbf{Y} \in \mathbb{R}^{d_1\times d_2}$ the observation matrix. In particular, we observe each elements independently with probability $p \in (0, 1)$. We denote $\Omega \subseteq [d_1] \times [d_2]$ by the index set of the observed entries, then $F_\Omega(\mathbf{U}, \mathbf{V})$ for matrix completion is formulated as $F_\Omega(\mathbf{U}, \mathbf{V}) = n^{-1}\sum_{i=1}^n F_{\Omega_{\mathcal{S}_i}}(\mathbf{U}, \mathbf{V})$, where each component function is defined as $F_{\Omega_{\mathcal{S}_i}}(\mathbf{U}, \mathbf{V}) = \sum_{(j,k)\in\Omega_{\mathcal{S}_i}} (\mathbf{U}_{j*}\mathbf{V}_{k*}^\top - Y_{jk})^2/(2b) + \mathcal{R}(\mathbf{U}, \mathbf{V})$. Note that $\{\Omega_{\mathcal{S}_i}\}_{i=1}^n$ denote the mutually disjoint subsets such that $\cup_{i=1}^n \Omega_{\mathcal{S}_i} = \Omega$. In addition, we have $|\Omega_{\mathcal{S}_i}| = b$ for $i = 1, \ldots, n$ such that $|\Omega| = nb$.

3. Main Theory

In this section, we present our main theoretical results for Algorithms 1 and 2. We first introduce several definitions for simplicity. Recall that the singular value decomposition of $\mathbf{X}^*$ is $\mathbf{X}^* = \overline{\mathbf{U}}^*\mathbf{\Sigma}^*\overline{\mathbf{V}}^{*\top}$, then following Tu et al. (2015); Zheng & Lafferty (2016), we define $\mathbf{Y}^* \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$ as the corresponding lifted positive semidefinite matrix of $\mathbf{X}^* \in \mathbb{R}^{d_1\times d_2}$ in higher dimension

$$\mathbf{Y}^* = \begin{bmatrix} \mathbf{U}^*\mathbf{U}^{*\top} & \mathbf{U}^*\mathbf{V}^{*\top} \\ \mathbf{V}^*\mathbf{U}^{*\top} & \mathbf{V}^*\mathbf{V}^{*\top} \end{bmatrix} = \mathbf{Z}^*\mathbf{Z}^{*\top},$$

where $\mathbf{U}^* = \overline{\mathbf{U}}^*(\mathbf{\Sigma}^*)^{1/2}$, $\mathbf{V}^* = \overline{\mathbf{V}}^*(\mathbf{\Sigma}^*)^{1/2}$, and $\mathbf{Z}^*$ is defined as $\mathbf{Z}^* = [\mathbf{U}^*; \mathbf{V}^*] \in \mathbb{R}^{(d_1+d_2)\times r}$. Besides, we define the solution set in terms of the true parameter $\mathbf{Z}^*$ as follows:

$$\mathcal{Z} = \Big\{ \mathbf{Z} \in \mathbb{R}^{(d_1+d_2)\times r} \,\Big|\, \mathbf{Z} = \mathbf{Z}^*\mathbf{R} \text{ for some } \mathbf{R} \in \mathbb{Q}_r \Big\},$$

where $\mathbb{Q}_r$ denotes the set of $r \times r$ orthonormal matrices. According to this definition, for any $\mathbf{Z} \in \mathcal{Z}$, we can obtain $\mathbf{X}^* = \mathbf{Z}_U\mathbf{Z}_V^\top$, where $\mathbf{Z}_U$ and $\mathbf{Z}_V$ denote the top $d_1 \times r$ and bottom $d_2 \times r$ matrices of $\mathbf{Z} \in \mathbb{R}^{(d_1+d_2)\times r}$ respectively.

Definition 3.1. Define the distance between $\mathbf{Z}$ and $\mathbf{Z}^*$ in terms of the optimal rotation as $d(\mathbf{Z}, \mathbf{Z}^*)$ such that

$$d(\mathbf{Z}, \mathbf{Z}^*) = \min_{\widetilde{\mathbf{Z}}\in\mathcal{Z}} \|\mathbf{Z} - \widetilde{\mathbf{Z}}\|_F = \min_{\mathbf{R}\in\mathbb{Q}_r} \|\mathbf{Z} - \mathbf{Z}^*\mathbf{R}\|_F.$$

Note that if $d(\mathbf{Z}, \mathbf{Z}^*) \leq \sqrt{\sigma_1}$, we have $\|\mathbf{X} - \mathbf{X}^*\|_F \leq c\sqrt{\sigma_1}d(\mathbf{Z}, \mathbf{Z}^*)$, where c is a constant (Yi et al., 2016).

Definition 3.2. Define the neighbourhood of $\mathbf{Z}^*$ with radius R as

$$\mathbb{B}(R) = \Big\{ \mathbf{Z} \in \mathbb{R}^{(d_1+d_2)\times r} \,\Big|\, d(\mathbf{Z}, \mathbf{Z}^*) \leq R \Big\}.$$

Next, we lay out several conditions, which are essential for proving our main theory. We impose restricted strong convexity (RSC) and smoothness (RSS) conditions (Negahban et al., 2009; Loh & Wainwright, 2013) on the sample loss function $\mathcal{L}_N$.

Condition 3.3 (Restricted Strong Convexity)**.** Assume $\mathcal{L}_N$ is restricted strongly convex with parameter μ, such that for all matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d_1\times d_2}$ with rank at most $3r$

$$\mathcal{L}_N(\mathbf{Y}) \geq \mathcal{L}_N(\mathbf{X}) + \langle \nabla\mathcal{L}_N(\mathbf{X}), \mathbf{Y} - \mathbf{X}\rangle + \frac{\mu}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2.$$

Condition 3.4 (Restricted Strong Smoothness)**.** Assume $\mathcal{L}_N$ is restricted strongly smooth with parameter L, such that for all matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d_1 \times d_2}$ with rank at most $3r$

$$\mathcal{L}_N(\mathbf{Y}) \leq \mathcal{L}_N(\mathbf{X}) + \langle \nabla \mathcal{L}_N(\mathbf{X}), \mathbf{Y} - \mathbf{X} \rangle + \frac{L}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2.$$

Based on Conditions 3.3 and 3.4, we prove that the sample loss function $\mathcal{L}_N$ satisfies a projected notion of the restricted Lipschitz continuous gradient property as displayed in the following lemma.

Lemma 3.5. Suppose the sample loss function $\mathcal{L}_N$ satisfies Conditions 3.3 and 3.4. For any rank-r matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d_1 \times d_2}$, let the singular value decomposition of $\mathbf{X}$ be $\overline{\mathbf{U}}_1 \mathbf{\Sigma}_1 \overline{\mathbf{V}}_1^\top$, then we have

$$\begin{aligned}\mathcal{L}_N(\mathbf{X}) \geq \mathcal{L}_N(\mathbf{Y}) &+ \langle \nabla \mathcal{L}_N(\mathbf{Y}), \mathbf{X} - \mathbf{Y} \rangle \\ &+ \frac{1}{4L}\|\widetilde{\mathbf{U}}^\top(\nabla \mathcal{L}_N(\mathbf{X}) - \nabla \mathcal{L}_N(\mathbf{Y}))\|_F^2 \\ &+ \frac{1}{4L}\|(\nabla \mathcal{L}_N(\mathbf{X}) - \nabla \mathcal{L}_N(\mathbf{Y}))\widetilde{\mathbf{V}}\|_F^2\big),\end{aligned}$$

where $\widetilde{\mathbf{U}} \in \mathbb{R}^{d_1 \times r_1}$ is an orthonormal matrix with $r_1 \leq 3r$ which satisfies $\mathrm{col}(\overline{\mathbf{U}}_1) \subseteq \mathrm{col}(\widetilde{\mathbf{U}})$, and $\widetilde{\mathbf{V}} \in \mathbb{R}^{d_2 \times r_2}$ is an orthonormal matrix with $r_2 \leq 3r$ that satisfies $\mathrm{col}(\overline{\mathbf{V}}_1) \subseteq \mathrm{col}(\widetilde{\mathbf{V}})$, and L is the RSS parameter.

Lemma 3.5 is essential to analyze the nonconvex optimization for low-rank matrix recovery and derive a linear convergence rate. Since the RSC and RSS conditions can only be verified over the subspace of low-rank matrices, the standard Lipschitz continuous gradient property could not be derived. That is why we need such a *restricted* version of Lipschitz continuous gradient property. To the best of our knowledge, this new notion of Lipschitz continuous gradient has never been proposed in the literature before. We believe it can be of broader interests for other nonconvex optimization problems to prove tight bounds.

Moreover, we assume that the gradient of the sample loss function $\nabla \mathcal{L}_N$ at $\mathbf{X}^*$ is upper bounded.

Condition 3.6. Recall the unknown rank-r matrix $\mathbf{X}^* \in \mathbb{R}^{d_1 \times d_2}$. Given a fixed sample size N and tolerance parameter $\delta \in (0, 1)$, we let $\epsilon(N, \delta)$ be the smallest scalar such that with probability at least $1 - \delta$, we have

$$\|\nabla \mathcal{L}_N(\mathbf{X}^*)\|_2 \leq \epsilon(N, \delta),$$

where $\epsilon(N, \delta)$ depends on sample size N and δ.

Finally, we assume that each component loss function $\mathcal{L}_i$ in (2.4) satisfies the restricted strong smoothness condition.

Condition 3.7 (Restricted Strong Smoothness for each Component)**.** Given a fixed batch size b, assume $\mathcal{L}_i$ is restricted strongly smooth with parameter L', such that for all matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{d_1 \times d_2}$ with rank at most $3r$

$$\mathcal{L}_i(\mathbf{Y}) \leq \mathcal{L}_i(\mathbf{X}) + \langle \nabla \mathcal{L}_i(\mathbf{X}), \mathbf{Y} - \mathbf{X} \rangle + \frac{L'}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2.$$

In latter analysis for generic setting, we assume that Conditions 3.3-3.7 hold, while for each specific model, we will verify these conditions respectively in the appendix.

3.1. Results for the Generic Setting

The following theorem shows that, in general, Algorithm 1 converges linearly to the unknown low-rank matrix $\mathbf{X}^*$ up to a statistical precision.

Theorem 3.8 (LRSVRG)**.** Suppose that Conditions 3.3, 3.4, 3.6, and 3.7 are satisfied. There exist constants c_1, c_2, c_3 and c_4 such that for any $\widetilde{\mathbf{Z}}^0 = [\widetilde{\mathbf{U}}^0; \widetilde{\mathbf{V}}^0] \in \mathbb{B}(c_2\sqrt{\sigma_r})$ with $c_2 \leq \min\{1/4, \sqrt{2\mu'/(5(3L+1))}\}$, if the sample size N is large enough such that $\epsilon^2(N, \delta) \leq c_2^2(1-\rho)\mu'\sigma_r^2/(c_3 r)$, where $\mu' = \min\{\mu, 1\}$, and the contraction parameter ρ is defined as follows:

$$\rho = \frac{10\kappa}{\mu'}\bigg(\frac{1}{\eta m \sigma_1} + c_4 \eta \sigma_1 L'^2\bigg),$$

then with the step size $\eta = c_1/\sigma_1$ and the number of iterations m properly chosen, the estimator $\widetilde{\mathbf{Z}}^S = [\widetilde{\mathbf{U}}^S; \widetilde{\mathbf{V}}^S]$ outputed from Algorithm 1 satisfies

$$\mathbb{E}\big[d^2(\widetilde{\mathbf{Z}}^S, \mathbf{Z}^*)\big] \leq \rho^S d^2(\widetilde{\mathbf{Z}}^0, \mathbf{Z}^*) + \frac{c_3 r \epsilon^2(N, \delta)}{(1-\rho)\mu'\sigma_r}, \quad (3.1)$$

with probability at least $1 - \delta$.

Remark 3.9. Theorem 3.8 implies that to achieve linear rate of convergence, it is necessary to set the step size η to be small enough and the inner loop iterations m to be large enough such that $\rho < 1$. Here we present a specific example to demonstrate such ρ is attainable. As stated in Theorem 3.8, if we set the step size $\eta = c_1'/\sigma_1$, where $c_1' = \mu'/\big(15 c_4 \kappa L'^2\big)$, then the contraction parameter ρ is calculated as follows:

$$\rho = \frac{10\kappa}{m\mu'\eta\sigma_1} + \frac{2}{3}.$$

Therefore, under the condition that $m \geq c_5\kappa^2$, we obtain $\rho \leq 5/6 < 1$, which leads to the linear convergence rate of Algorithm 1. Besides, our algorithm also achieves the linear convergence in terms of reconstruction error, since the reconstruction error $\|\widetilde{\mathbf{X}}^s - \mathbf{X}^*\|_F^2$ can be upper bounded by $C\sigma_1 \cdot d^2(\widetilde{\mathbf{Z}}^s, \mathbf{Z}^*)$, where C is a constant.

Remark 3.10. The right hand side of (3.1) consists of two parts, where the first one represents the optimization error and the second one denotes the statistical error. Note that in the noiseless case, since $\epsilon(N, \delta) = 0$, the statistical error becomes zero. As stated in Remark 3.9, with

appropriate η and m, we are able to achieve the linear rate of convergence. Therefore, in order to make sure the optimization error satisfies $\rho^S d^2(\widetilde{\mathbf{Z}}^0, \mathbf{Z}^*) \leq \epsilon$, it suffices to perform $S = O\big(\log(1/\epsilon)\big)$ outer loop iterations. Recall that from Remark 3.9 we have $m = O(\kappa^2)$. Since for each outer loop iteration, it is required to calculate m mixed stochastic variance-reduced gradients and one full gradient, the overall gradient complexity for our algorithm to achieve ϵ precision is

$$O\Big((N + \kappa^2 b)\log\Big(\frac{1}{\epsilon}\Big)\Big).$$

However, the gradient complexity of the state-of-the-art gradient descent based algorithm (Wang et al., 2016) to achieve ϵ precision is $O\big(N\kappa\log(1/\epsilon)\big)$. Therefore, provided that $\kappa \leq n$, our method is computationally more efficient than the state-of-the-art gradient descent approach. The detailed comparison of the overall computational complexity among different methods for each specific model can be found in next subsection.

To satisfy the initial condition $\widetilde{\mathbf{Z}}^0 \in \mathbb{B}(c_2\sqrt{\sigma_r})$ in Theorem 3.8, according to Lemma 5.14 in Tu et al. (2015), it suffices to guarantee that $\widetilde{\mathbf{X}}^0$ is close enough to the unknown rank-r matrix $\mathbf{X}^*$ such that $\|\widetilde{\mathbf{X}}^0 - \mathbf{X}^*\|_F \leq c\sigma_r$, where $c \leq \min\{1/2, 2c_2\}$. The following theorem shows the output of Algorithm 2 can satisfy this condition.

Theorem 3.11. (Wang et al., 2016) Suppose the sample loss function $\mathcal{L}_N$ satisfies Conditions 3.3, 3.4 and 3.6. Let $\widetilde{\mathbf{X}}^0 = \widetilde{\mathbf{U}}^0\widetilde{\mathbf{V}}^{0\top}$, where $(\widetilde{\mathbf{U}}^0, \widetilde{\mathbf{V}}^0)$ is the produced initial solution in Algorithm 2. If $L/\mu \in (1, 4/3)$, then with step size $\tau = 1/L$, we have with probability at least $1 - \delta$ that

$$\|\widetilde{\mathbf{X}}^0 - \mathbf{X}^*\|_F \leq \rho^T\|\mathbf{X}^*\|_F + \frac{2\sqrt{3r}\epsilon(N,\delta)}{L(1-\rho)},$$

where $\rho = 2\sqrt{1-\mu/L}$ is the contraction parameter.

Theorem 3.11 suggests that, in order to guarantee $\|\widetilde{\mathbf{X}}^0 - \mathbf{X}^*\|_F \leq c\sigma_r$, we need to perform at least $T = \log(c'\sigma_r/\|\mathbf{X}^*\|_F)/\log(\rho)$ number of iterations to ensure the optimization error is small enough, and it is also necessary to make sure the sample size N is large enough such that $\epsilon(N,\delta) \leq c'L(1-\rho)\sigma_r/\big(2\sqrt{3r}\big)$, which corresponds to a sufficiently small statistical error.

3.2. Implications for Specific Models

In this subsection, we demonstrate the implications of our generic theory to specific models. For each specific model, we only need to verify Conditions 3.3-3.7. We denote $d = \max\{d_1, d_2\}$ in the following discussions.

3.2.1. Matrix Sensing

We provide the theoretical guarantee of our algorithm for matrix sensing.

Corollary 3.12. Consider matrix sensing with standard normal linear operator $\mathcal{A}$ and noise vector $\boldsymbol{\epsilon}$, whose entries follow i.i.d. sub-Gaussian distribution with parameter ν. There exist constants $\{c_i\}_{i=1}^8$ such that if the number of observations satisfies $N \geq c_1 rd$ and we choose the parameters $\eta = c_2/\sigma_1$, where $c_2 = \mu'/\big(c_3\kappa\big)$, $m \geq c_4\kappa^2$, then for any initial solution satisfies $\widetilde{\mathbf{Z}}^0 \in \mathbb{B}(c_5\sqrt{\sigma_r})$, with probability at least $1 - c_6\exp\big(-c_7 d\big)$, the output of Algorithm 1 satisfies

$$\mathbb{E}\big[d^2(\widetilde{\mathbf{Z}}^S, \mathbf{Z}^*)\big] \leq \rho^S d^2(\widetilde{\mathbf{Z}}^0, \mathbf{Z}^*) + c_8\nu^2\frac{rd}{N}, \tag{3.2}$$

where the contraction parameter $\rho < 1$.

Remark 3.13. According to (3.2), in the noisy setting, the output of our algorithm achieves $O\big(\sqrt{rd/N}\big)$ statistical error after $O\big(\log(N/(rd))\big)$ number of outer loop iterations. This statistical error matches the minimax lower bound for matrix sensing (Negahban & Wainwright, 2011). In the noiseless case, to ensure the restricted strong convexity and smoothness conditions of our objective function, we require sample size $N = O(rd)$, which attains the optimal sample complexity for matrix sensing (Recht et al., 2010; Tu et al., 2015; Wang et al., 2016). Most importantly, from Remark 3.10 we know that for the output $\widetilde{\mathbf{Z}}^S$ of our algorithm, the overall computational complexity of our algorithm to achieve ϵ precision for matrix sensing is $O\big((Nd^2 + \kappa^2 bd^2)\log(1/\epsilon)\big)$. Nevertheless, the overall computational complexity for the state-of-the-art gradient descent algorithms in both noiseless (Tu et al., 2015) and noisy (Wang et al., 2016) cases to obtain ϵ precision is $O\big(N\kappa d^2\log(1/\epsilon)\big)$. Therefore, our algorithm is more efficient provided that $\kappa \leq n$, which is consistent with the result obtained by (Zhang et al., 2017b). In their work, they proposed an accelerated stochastic gradient descent method for matrix sensing based on the restricted isometry property. However, since the restricted isometry property is more restrictive than the restricted strong convex and smoothness conditions, their results cannot be applied to more general low-rank matrix recovery problems.

3.2.2. Matrix Completion

We provide the theoretical guarantee of our algorithm for matrix completion. In particular, we consider a partial observation model, which means only the elements over a subset $\mathcal{X} \subseteq [d_1] \times [d_2]$ are observed. In addition, we assume a uniform sampling model for $\mathcal{X}$, which is defined as $\forall(j,k) \in \mathcal{X}$, $j \sim$ uniform$([d_1])$, $k \sim$ uniform$([d_2])$. To avoid overly sparse matrices (Gross, 2011; Negahban & Wainwright, 2012), we impose the following incoherence condition (Candès & Recht, 2009). More specifically, suppose the singular value decomposition of $\mathbf{X}^*$ is $\mathbf{X}^* = \overline{\mathbf{U}}^*\boldsymbol{\Sigma}^*\overline{\mathbf{V}}^{*\top}$, we assume the following conditions hold $\|\overline{\mathbf{U}}^*\|_{2,\infty} \leq \sqrt{\beta r/d_1}$ and $\|\overline{\mathbf{V}}^*\|_{2,\infty} \leq$

$\sqrt{\beta r/d_2}$, where r denotes the rank of $\mathbf{X}^*$, and β denotes the incoherence parameter for $\mathbf{X}^*$.

In order to make sure our produced estimator satisfies incoherence constraint, we need a projection step, which is displayed in Algorithm 1. Therefore, we construct two feasible sets $\mathcal{C}_i = \{\mathbf{A} \in \mathbb{R}^{d_i \times r} \,\big|\, \|\mathbf{A}\|_{2,\infty} \leq \alpha_i\}$, where $\alpha_i = \sqrt{\beta r \sigma_1 / d_i}$, and $i \in \{1, 2\}$. Thus for any $\mathbf{U} \in \mathcal{C}_1$ and $\mathbf{V} \in \mathcal{C}_2$, we have $\mathbf{X} = \mathbf{U}\mathbf{V}^\top \in \mathcal{C} = \{\mathbf{A} \in \mathbb{R}^{d_1 \times d_2} \,\big|\, \|\mathbf{A}\|_{\infty,\infty} \leq \alpha\}$, where $\alpha = \beta r \sigma_1 / \sqrt{d_1 d_2}$.

We have the following convergence result of our algorithm for matrix completion.

Corollary 3.14. Consider noisy matrix completion under uniform sampling model. Suppose $\mathbf{X}^*$ satisfies incoherence condition. There exist constants $\{c_i\}_{i=1}^7$ such that if we choose parameters $\eta = c_1/\sigma_1$, where $c_1 = \mu'/\big(c_2\kappa\big)$, $m \geq c_3\kappa^2$, and the number of observations satisfies $N \geq c_4 r^2 d \log d$, then for any initial solution satisfies $\widetilde{\mathbf{Z}}^0 \in \mathbb{B}(c_5\sqrt{\sigma_r})$, then with probability at least $1 - c_6/d$, the output of Algorithm 1 satisfies

$$\mathbb{E}[d^2(\widetilde{\mathbf{Z}}^S, \mathbf{Z}^*)] \leq \rho^S d^2(\widetilde{\mathbf{Z}}^0, \mathbf{Z}^*) + c_7 \Gamma \frac{r d \log d}{N}, \quad (3.3)$$

where $\Gamma = \max\{\nu^2, r\beta^2\sigma_1^2\}$, the contraction parameter $\rho < 1$.

Remark 3.15. Corollary 3.14 implies that after $O\big(\log(N/(r^2 d \log d))\big)$ number of outer loops, our algorithm achieves $O\big(r\sqrt{d \log d / N}\big)$ statistical error, which is near optimal compared with the minimax lower bound $O(\sqrt{rd \log d / N})$ for matrix completion proved in Negahban & Wainwright (2012); Koltchinskii et al. (2011). And its sample complexity is $O(r^2 d \log d)$, which matches the best-known sample complexity of matrix completion using nonconvex matrix factorization (Zheng & Lafferty, 2016). Recall that from Remark 3.10, the overall computational complexity of our algorithm to reach ϵ accuracy for matrix completion is $O\big((N + \kappa^2 b) r^3 d \log(1/\epsilon)\big)$. However, for the state-of-the-art gradient descent based algorithms, the computational complexity for both noiseless (Zheng & Lafferty, 2016) and noisy (Wang et al., 2016) cases to obtain ϵ accuracy is $O\big(N\kappa r^3 d \log(1/\epsilon)\big)$. Thus the computational complexity of our algorithm is lower than the state-of-the-art gradient descent methods if we have $\kappa \leq n$. In addition, for the online stochastic gradient descent algorithm (Jin et al., 2016), the overall computational complexity is $O(r^4\kappa^4 d \log(1/\epsilon))$. Since their results has a fourth power dependency on both r and κ, our method can yield a significant improvement over the online method when r, κ is large.

4. Experiments

In this section, we present the experimental performance of our proposed algorithm for different models based on numerical simulations and real data experiments.

4.1. Numerical Simulations

We first investigate the effectiveness of our proposed algorithm compared with the state-of-the-art gradient descent algorithm (Wang et al., 2016; Zheng & Lafferty, 2016). Then, we evaluate the sample complexity required by both methods to achieve exact recovery in the noiseless case. Finally, we illustrate the statistical error of our method in the noisy case. Note that both algorithms use the same initialization method (Algorithm 2) with optimal parameters selected by cross validation. Furthermore, all results are averaged over 30 trials. Note that due to the space limit, we only lay out simulation results for matrix completion, results for other models can be found in Appendix A.

For matrix completion, we consider the unknown low-rank matrix $\mathbf{X}^*$ in the following settings: (i) $d_1 = 100, d_2 = 80, r = 2$; (ii) $d_1 = 120, d_2 = 100, r = 3$; (iii) $d_1 = 140, d_2 = 120, r = 4$. First, we generate the unknown low-rank matrix $\mathbf{X}^*$ as $\mathbf{X}^* = \mathbf{U}^*\mathbf{V}^{*\top}$, where $\mathbf{U}^* \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{V}^* \in \mathbb{R}^{d_2 \times r}$ are randomly generated. Next, we use uniform observation model to obtain data matrix $\mathbf{Y}$. Finally, we consider two settings: (1) noisy case: the noise follows i.i.d. normal distribution with variance $\sigma^2 = 0.25$ and (2) noiseless case.

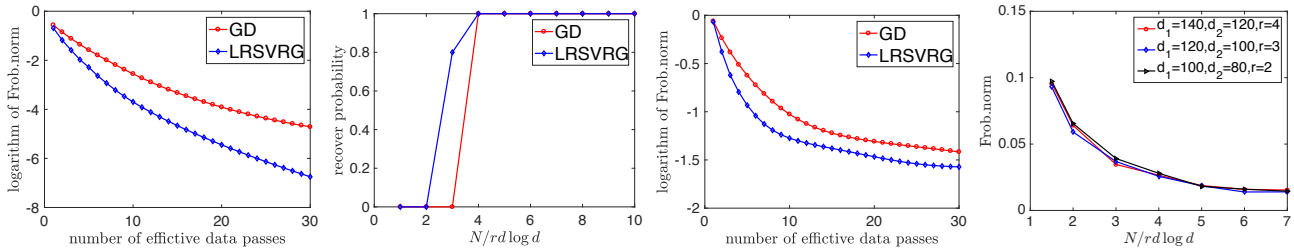
For the results of convergence rate, we show the mean squared error $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F^2/(d_1 d_2)$ in log scale versus number of effective data passes. Figures 1(a) and 1(c) illustrate the linear rate of convergence of our algorithm (LRSVRG) in the setting (i). The results imply that after the same number of effective data passes, our algorithm is more efficient than the state-of-the-art gradient descent algorithm in estimation error. For the results of sample complexity, we illustrate the empirical probability of exact recovery under rescaled sample size $N/(rd \log d)$. For the estimator $\widehat{\mathbf{X}}$ given by different algorithms, it is considered to achieve exact recovery, if the relative error $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F / \|\mathbf{X}^*\|_F$ is less than 10^{-3}. Figure 1(b) shows the empirical recovery probability of different methods in the setting (i). It implies a phase transition around $N = 3rd \log d$. Although our theoretical results requires $O(r^2 d \log d)$ sample complexity, the simulation results suggest that our method achieves the optimal sample complexity $N = O(rd \log d)$. Note that we leave out results in other settings to avoid redundancy since we get similar patterns for these results. The results of statistical error are displayed in Figure 1(d), which is consistent with our main result in Corollary 3.14.

4.2. Real Data Experiments

We apply our proposed stochastic variance-reduced gradient algorithm for matrix completion to collaborative filtering in recommendation system, and compare it with sev-

Table 1. Experimental results of collaborative filtering in terms of averaged RMSE and CPU time for different algorithms.

Dataset	Performance	SVP	SOFTIMPUTE	ALTMIN	TNC	RIMP	NUCLEAR	SCAD	GD	LRSVRG
JESTER1	RMSE	4.7318	5.1211	4.8562	4.4803	4.3401	4.6910	4.1733	4.1832	4.1605
	Time (s)	18.71	161.08	11.55	29.63	1.92	192.15	197.52	1.01	0.81
JESTER2	RMSE	4.7712	5.1523	4.8712	4.4511	4.3721	4.5597	4.2016	4.2177	4.1909
	Time (s)	16.94	152.82	10.68	28.81	1.75	166.94	171.31	0.96	0.71
JESTER3	RMSE	8.7439	5.4532	9.5230	4.6712	4.9803	5.1231	4.6777	4.6867	4.6247
	Time (s)	16.69	10.82	12.57	12.84	0.95	94.88	253.73	0.86	0.54



(a) Noiseless Case (b) Sample Complexity (c) Noisy Case (d) Statistical Error

Figure 1. Numerical results for matrix completion. (a) and (c) Rate of convergence for matrix completion in the noiseless and noisy case, respectively: logarithm of mean squared error $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F^2/(d_1 d_2)$ versus number of effective data passes, which demonstrates the effectiveness of our method; (b) Empirical probability of exact recovery versus $N/(rd\log d)$; (d) Statistical error for matrix completion: mean squared error in Frobenius norm $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F^2/(d_1 d_2)$ versus rescaled sample size $N/(rd\log d)$.

eral state-of-the-art matrix completion algorithms, including singular value projection (SVP) (Jain et al., 2010), trace norm constraint (TNC) (Jaggi et al., 2010), alternating minimization (AltMin) (Jain et al., 2013b), spectral regularization (SoftImpute) (Mazumder et al., 2010), rank-one matrix pursuit (Wang et al., 2014), nuclear norm penalty (Negahban & Wainwright, 2011), nonconvex SCAD penalty (Gui & Gu, 2015) and gradient descent (Zheng & Lafferty, 2016). In particular, we use three large recommendation datasets called Jester1, Jester2 and Jester3 (Goldberg et al., 2001), which contain anonymous ratings on 100 jokes from different users. The jester datasets consist of $\{24983, 23500, 24938\}$ rows and 100 columns respectively, with $\{10^6, 10^6, 6 \times 10^5\}$ ratings correspondingly. Besides, the rating scales take value from $[-10, 10]$. Our goal is to recover the whole rating matrix based on partial observations. Therefore, we randomly choose half of the ratings as our observed data, and predict the other half based on different matrix completion algorithms. We perform 10 different observed/unobserved entry splittings, and record the averaged root mean square error (RMSE) as well as CPU time for different algorithms. We summarize the comparisons in Table 1, which suggests that our proposed LRSVRG algorithm outperforms all the other baseline algorithms in terms of RMSE and CPU time, which aligns well with our theory.

5. Conclusions and Future Work

We proposed a unified stochastic variance-reduced gradient descent framework for low-rank matrix recovery that integrates both optimization-theoretic and statistical analyses. Based on the mild restricted strong convexity and smoothness conditions, we derived a projected notion of the restricted Lipschitz continuous gradient property, and established the linear convergence rate of our proposed algorithm. With an appropriate initialization procedure, we proved that our algorithm enjoys improved computational complexity compared with existing approaches. There are still many interesting problems along this line of research. For example, we will study accelerating the low-rank plus sparse matrix/tensor recovery (Gu et al., 2014; 2016; Yi et al., 2016; Zhang et al., 2017a) through variance reduction technique in the future.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1618948 and IIS-1652539. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Agarwal, Alekh, Negahban, Sahand, and Wainwright, Martin J. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pp. 37–45, 2010.

Allen-Zhu, Zeyuan and Hazan, Elad. Variance reduction for faster non-convex optimization. *arXiv preprint arXiv:1603.05643*, 2016.

Bhaskar, Sonia A and Javanmard, Adel. 1-bit matrix completion under exact low-rank constraint. In *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, pp. 1–6. IEEE, 2015.

Bhojanapalli, Srinadh, Kyrillidis, Anastasios, and Sanghavi, Sujay. Dropping convexity for faster semi-definite optimization. *arXiv preprint*, 2015.

Cabral, Ricardo Silveira, De la Torre, Fernando, Costeira, João Paulo, and Bernardino, Alexandre. Matrix completion for multi-label image classification. In *NIPS*, volume 201, pp. 2, 2011.

Cai, Tony and Zhou, Wen-Xin. A max-norm constrained minimization approach to 1-bit matrix completion. *Journal of Machine Learning Research*, 14(1):3619–3647, 2013.

Candès, Emmanuel J and Recht, Benjamin. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Candès, Emmanuel J and Tao, Terence. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

Chen, Jinghui and Gu, Quanquan. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

Chen, Yudong and Wainwright, Martin J. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Davenport, Mark A, Plan, Yaniv, van den Berg, Ewout, and Wootters, Mary. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.

De Sa, Christopher, Olukotun, Kunle, and Ré, Christopher. Global convergence of stochastic gradient descent for some non-convex matrix problems. *arXiv preprint arXiv:1411.1134*, 2014.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014a.

Defazio, Aaron J, Caetano, Tibério S, and Domke, Justin. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conference on Machine Learning*, 2014b.

Goldberg, Ken, Roeder, Theresa, Gupta, Dhruv, and Perkins, Chris. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

Gross, David. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

Gu, Quanquan, Gui, Huan, and Han, Jiawei. Robust tensor decomposition with gross corruption. In *Advances in Neural Information Processing Systems*, pp. 1422–1430, 2014.

Gu, Quanquan, Wang, Zhaoran Wang, and Liu, Han. Low-rank and sparse structure pursuit via alternating minimization. In *Artificial Intelligence and Statistics*, pp. 600–609, 2016.

Gui, Huan and Gu, Quanquan. Towards faster rates and oracle property for low-rank matrix estimation. *arXiv preprint arXiv:1505.04780*, 2015.

Hardt, Moritz. Understanding alternating minimization for matrix completion. In *FOCS*, pp. 651–660. IEEE, 2014.

Hardt, Moritz and Wootters, Mary. Fast matrix completion without the condition number. In *COLT*, pp. 638–678, 2014.

Hardt, Moritz, Meka, Raghu, Raghavendra, Prasad, and Weitz, Benjamin. Computational limits for matrix completion. In *COLT*, pp. 703–725, 2014.

Jaggi, Martin, Sulovsk, Marek, et al. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 471–478, 2010.

Jain, Prateek and Netrapalli, Praneeth. Fast exact matrix completion with finite samples. *arXiv preprint*, 2014.

Jain, Prateek, Meka, Raghu, and Dhillon, Inderjit S. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pp. 937–945, 2010.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *STOC*, pp. 665–674, 2013a.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674. ACM, 2013b.

Jin, Chi, Kakade, Sham M, and Netrapalli, Praneeth. Provable efficient online matrix completion via non-convex stochastic gradient descent. *arXiv preprint arXiv:1605.08370*, 2016.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Keshavan, Raghunandan H, Oh, Sewoong, and Montanari, Andrea. Matrix completion from a few entries. In *2009 IEEE International Symposium on Information Theory*, pp. 324–328. IEEE, 2009.

Keshavan, Raghunandan H, Montanari, Andrea, and Oh, Sewoong. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010.

Koltchinskii, Vladimir, Lounici, Karim, Tsybakov, Alexandre B, et al. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5): 2302–2329, 2011.

Konečnỳ, Jakub and Richtárik, Peter. Semi-stochastic gradient descent methods. *arXiv:1312.1666*, 2013.

Konečnỳ, Jakub, Liu, Jie, Richtárik, Peter, and Takáč, Martin. ms2gd: Mini-batch semi-stochastic gradient descent in the proximal setting. *arXiv:1410.4744*, 2014.

Loh, Po-Ling and Wainwright, Martin J. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pp. 476–484, 2013.

Mairal, Julien. Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv:1402.4419*, 2014.

Mazumder, Rahul, Hastie, Trevor, and Tibshirani, Robert. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

Negahban, Sahand and Wainwright, Martin J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, pp. 1069–1097, 2011.

Negahban, Sahand and Wainwright, Martin J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May): 1665–1697, 2012.

Negahban, Sahand, Yu, Bin, Wainwright, Martin J, and Ravikumar, Pradeep K. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pp. 1348–1356, 2009.

Ni, Renkun and Gu, Quanquan. Optimal statistical and computational rates for one bit matrix completion. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 426–434, 2016.

Park, Dohyung, Kyrillidis, Anastasios, Bhojanapalli, Srinadh, Caramanis, Constantine, and Sanghavi, Sujay. Provable burer-monteiro factorization for a class of norm-constrained matrix problems. *stat*, 1050:1, 2016a.

Park, Dohyung, Kyrillidis, Anastasios, Caramanis, Constantine, and Sanghavi, Sujay. Finding low-rank solutions to matrix problems, efficiently and provably. *arXiv preprint arXiv:1606.03168*, 2016b.

Recht, Benjamin, Fazel, Maryam, and Parrilo, Pablo A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Reddi, Sashank J, Hefny, Ahmed, Sra, Suvrit, Póczós, Barnabás, and Smola, Alex. Stochastic variance reduction for nonconvex optimization. *arXiv preprint arXiv:1603.06160*, 2016.

Rennie, Jasson DM and Srebro, Nathan. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pp. 713–719. ACM, 2005.

Rohde, Angelika, Tsybakov, Alexandre B, et al. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011.

Schmidt, Mark, Roux, Nicolas Le, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

Srebro, Nathan, Rennie, Jason, and Jaakkola, Tommi S. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pp. 1329–1336, 2004.

Sun, Ruoyu and Luo, Zhi-Quan. Guaranteed matrix completion via nonconvex factorization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pp. 270–289. IEEE, 2015.

Tu, Stephen, Boczar, Ross, Soltanolkotabi, Mahdi, and Recht, Benjamin. Low-rank solutions of linear matrix equations via procrustes flow. *arXiv preprint arXiv:1507.03566*, 2015.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Wang, Lingxiao, Zhang, Xiao, and Gu, Quanquan. A unified computational and statistical framework for nonconvex low-rank matrix estimation. *arXiv preprint arXiv:1610.05275*, 2016.

Wang, Zheng, Lai, Ming-Jun, Lu, Zhaosong, Fan, Wei, Davulcu, Hasan, and Ye, Jieping. Rank-one matrix pursuit for matrix completion. In *ICML*, pp. 91–99, 2014.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Xu, Miao, Jin, Rong, and Zhou, Zhi-Hua. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems*, pp. 2301–2309, 2013.

Yi, Xinyang, Park, Dohyung, Chen, Yudong, and Caramanis, Constantine. Fast algorithms for robust pca via gradient descent. In *Advances in Neural Information Processing Systems*, pp. 4152–4160, 2016.

Zhang, Aston and Gu, Quanquan. Accelerated stochastic block coordinate descent with optimal sampling. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2035–2044. ACM, 2016.

Zhang, Xiao, Wang, Lingxiao, and Gu, Quanquan. A nonconvex free lunch for low-rank plus sparse matrix recovery. *arXiv preprint arXiv:1702.06525*, 2017a.

Zhang, Xiao, Wang, Lingxiao, and Gu, Quanquan. Stochastic variance-reduced gradient descent for low-rank matrix recovery from linear measurements. *arXiv preprint arXiv:1701.00481*, 2017b.

Zhao, Tuo, Wang, Zhaoran, and Liu, Han. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pp. 559–567, 2015.

Zheng, Qinqing and Lafferty, John. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pp. 109–117, 2015.

Zheng, Qinqing and Lafferty, John. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.