SIAM J. IMAGING SCIENCES
Vol. 11, No. 4, pp. 2165–2204

Finding Low-Rank Solutions via Nonconvex Matrix Factorization, Efficiently and Provably*

Dohyung Park†, Anastasios Kyrillidis‡, Constantine Caramanis§, and Sujay Sanghavi§

Abstract. A rank-r matrix $X \in \mathbb{R}^{m\times n}$ can be written as a product $UV^\top$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$. One could exploit this observation in optimization: e.g., consider the minimization of a convex function $f(X)$ over rank-r matrices, where the set of low-rank matrices is modeled via $UV^\top$. Though such parameterization reduces the number of variables and is more computationally efficient (of particular interest is the case $r \ll \min\{m,n\}$), it comes at a cost: $f(UV^\top)$ becomes a nonconvex function w.r.t. U and V. We study such parameterization on generic convex objectives f and focus on first-order, gradient descent algorithms. We propose the *bifactored gradient descent* (`BFGD`) algorithm, an efficient first-order method that operates directly on the U, V factors. We show that when f is (restricted) smooth, `BFGD` has local sublinear convergence; when f is both (restricted) smooth and (restricted) strongly convex, it has local linear convergence. For several applications, we provide simple and efficient initialization schemes that provide initial conditions, good enough for the above convergence results to hold, globally. Extensive experimental results support our arguments that `BFGD` is an efficient and accurate nonconvex method, compared to state-of-the-art approaches.

Key words. nonconvex optimization, matrix factorization, low-rank minimization

AMS subject classifications. 90C06, 90C26, 65K05

DOI. 10.1137/17M1150189

1. Introduction.

We study matrix problems of the form

$$\underset{X\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad f(X), \tag{1}$$

where the minimizer $X^\star \in \mathbb{R}^{m\times n}$ is rank-$r^\star$ ($r^\star \le \min\{m,n\}$) or *nearly* low rank; i.e., $\|X^\star - X^\star_{r^\star}\|_F$ is sufficiently small, for $X^\star_{r^\star}$ being the best rank-$r^\star$ approximation of $X^\star$. In our discussions, f is a differentiable convex function. Further assumptions on f will be described later in the text.

Specific instances of (1) appear in several applications in diverse research fields. A non-exhaustive list includes factorization-based recommender systems [88, 86, 35, 12, 63, 51, 59], multilabel classification tasks [3, 13, 27, 76, 96, 101], dimensionality reduction techniques

*Received by the editors October 2, 2017; accepted for publication (in revised form) July 13, 2018; published electronically October 2, 2018. A preliminary version of this paper appeared in *Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing*, 2016. The present draft is an expanded version containing additional results.

http://www.siam.org/journals/siims/11-4/M115018.html

†Facebook, Seattle, WA 98109 (dohyung22.park@gmail.com, http://dhpark22.github.io/).

‡IBM T.J. Watson Research Center, Yorktown Heights, NY 10548, and Rice University, Houston, TX 77005 (anastasios@rice.edu, http://akyrillidis.github.io/).

§Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 (constantine@utexas.edu, sanghavi@mail.utexas.edu).

[87, 31, 56, 94, 44, 74], density matrix estimation of quantum systems [1, 43, 58], phase retrieval applications [24, 95], sensor localization [16, 100] and protein clustering [75] tasks, image processing problems [5], as well as applications in system theory [41]. Thus, it is critical to devise user-friendly, efficient, and provable algorithms for (1), taking into consideration the (near) low-rank structure of $X^\star$.

In general, imposing a low-rank constraint could result in an NP-hard problem. However, (1) with a rank constraint can be solved in polynomial time for applications where f has specific structure. A prime example is the matrix sensing (MS) problem [26, 85, 51]; see section 1.1. There, $X^\star$ can be recovered in polynomial time by solving (1) with a rank constraint [53, 11, 8, 69, 65, 90] or by solving its convex nuclear-norm relaxation, as in [72, 10, 23, 9, 28, 104].

Although algorithms operating on X space have attractive convergence rates, they simultaneously manipulate $m \times n$ variables in X. This is computationally expensive in the high-dimensional regime: typically, each iteration requires computing at least the top-r singular value/vectors of matrices. As $\{m, n\}$ scale, the computational demands per iteration are prohibitive.

Optimizing over factors. In this paper, we follow a different path: a rank-r matrix $X \in \mathbb{R}^{m\times n}$ can be written as a product of two matrices $UV^\top$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$. Based on this, we are interested in (1) via the $UV^\top$ parametrization:

$$\underset{U\in\mathbb{R}^{m\times r},\ V\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad f(UV^\top), \qquad \text{where } r \le \text{rank}(X^\star) \le \{m,\ n\}. \tag{2}$$

Note that characterizations (2) and (1) are equivalent in the case $\text{rank}(X^\star) = r$.[1] Observe that such parameterization leads to a very specific kind of nonconvexity in f. Proving convergence for these settings becomes a harder task, due to the bilinearity of the variable space.

Motivation. When r is much smaller than $\min\{m, n\}$, $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ contain far fewer variables than $X = UV^\top$. Thus, by construction, such parametrization makes it easier to update and store the iterates U, V.

Key is that $UV^\top$ reformulation automatically encodes the rank constraint. Approaches working on X require computing a truncated SVD[2] per iteration, which can get cumbersome in large-scale settings. In stark contrast, working with $f(UV^\top)$ replaces singular value computations with matrix-matrix multiplication operations. See section 7.1 for some empirical evidence of the above.

Our contributions. Such bilinear reformulations $X = UV^\top$often lack theoretical guarantees. Only recently, there have been attempts in providing answers to when and why such nonconvex approaches perform well in theory; see [52, 4, 92, 107, 30, 14, 106, 89, 108, 55, 71, 97, 110, 98, 46, 45].

Our work is more general, supplements current state of the art, and further focuses on the practical aspects of such nonconvex problems. We address important issues in practice:

[1] By equivalent, we mean that the set of global minima in (2) contains that of (1). It remains an open question though whether the reformulation in (2) introduces spurious local minima in the factored space for *the majority of f cases.*

[2] This holds in the best scenario; in the convex case, where the rank constraint is "relaxed" by the nuclear norm, the projection onto the nuclear-norm ball often requires a full SVD calculation.

how to select initial points, how to set the step size, etc. At the same time, we back up our findings with theoretical results. Our contributions can be summarized as follows:

- We study gradient descent on (2) for *nonsquare* matrices. We call this *bifactored gradient descent* (BFGD). The current literature relies on special cases of f for theoretical results [89, 92, 108, 106]. In this work, *we propose a more generic perspective of such factorization techniques:* under general assumptions such as *smoothness and strong convexity of* f, our theory applies automatically and does not rely on any specific structures in f.
- When f is only (restricted) smooth, we show that a simple lifting technique leads to a local sublinear rate convergence guarantee, based on the positive semidefinite (PSD) results in [14]. With a more careful analysis, we improve upon [14] with a weaker initial condition.
- When f is both (restricted) strongly convex and smooth, results from the PSD case do not readily apply. Of significant importance is the use of a regularizer that restricts the geometry of the problem. Here, we improve upon [92, 108, 103]—where such a regularizer was used only for the cases of matrix sensing/completion and robust principal component analysis (PCA)—and solve a different formulation that leads to local linear rate convergence guarantees. *Our proof technique is a generalization to the current known theory*: using any smooth and strongly convex regularizer on the term $(U^\top U - V^\top V)$, with optimum at zero, one can guarantee locally linear convergence.
- Our theory is backed up with extensive experiments, including affine rank minimization (section 7.3), compressed noisy image reconstruction (section 7.4), and 1-bit matrix completion tasks (section 7.5). Our proposed scheme shows superior performance, as compared to state-of-the-art approaches, while being (i) simple to implement, (ii) scalable in practice, and (iii) versatile to various applications.

Our work is a nontrivial generalization of the PSD case [14]. This is because, for any $UV^\top$, we have $UV^\top = (\delta U)(\frac{1}{\delta}V)^\top$ for $\delta > 0$. This makes the analysis over the factors ill-conditioned. Such cases break down the existing theory on PSD matrices [14] and make necessary the utilization of balancing regularizers that ensure U, V estimates do not behave differently w.r.t. their spectrum energy.

1.1. When such optimization criteria appear in practice.

In this section, we briefly describe applications that can be modeled as in (2).

1.1.1. Matrix sensing applications.

MS [40, 85] involves the recovery of a *low-rank* $X^\star$ *from a limited set of linear measurements*, *i.e.*,

$$\underset{X\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad f(X) := \tfrac{1}{2}\cdot\|y-\mathcal{A}(X)\|_2^2 \quad \text{subject to} \quad \operatorname{rank}(X)\le r, \tag{3}$$

where usually $m \neq n$ and $r \ll \min\{m,\ n\}$. Here, $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ is a sensing map and $y = \mathcal{A}(X^\star) + \varepsilon \in \mathbb{R}^p$ contains the noisy samples, where $p \ll m \cdot n$. Critical assumption for $\mathcal{A}$ that renders (3) a polynomially solvable problem is the *restricted isometry property* (RIP) for low-rank matrices [25].

Definition 1.1 (restricted isometry property). *A linear map* $\mathcal{A}$ *satisfies the* r*-RIP with constant* δ_r *if*

$$(1-\delta_r)\|X\|_F^2 \le \|\mathcal{A}(X)\|_2^2 \le (1+\delta_r)\|X\|_F^2$$

is satisfied for all matrices $X \in \mathbb{R}^{n\times n}$ *such that rank*$(X) \le r$.

Linear maps that satisfy Definition 1.1 also satisfy (restricted) smoothness and strong convexity [81]; see Theorem 2 in [29] and section 2 for their definition.

State-of-the-art approaches. The most popularized approach for (3) is through *convexification* [39, 85, 26]:

$$\underset{X\in\mathbb{R}^{n\times p}}{\text{minimize}} \quad f(X) \quad \text{subject to} \quad \|X\|_* \le t, \tag{4}$$

where $\|X\|_*$ denotes the nuclear norm of X. Efficient implementations can be found in [72, 10, 23, 9]. However, due to the nuclear norm, these methods require full SVDs per iteration, which makes them impractical in large-scale settings. From a nonconvex perspective, algorithms that solve (3) in a nonfactored form include `SVP` and randomized `SVP` [53, 11], the Riemannian trust region matrix completion algorithm (`RTRMC`) [18], `ADMiRA` [69], and the `Matrix ALPS` framework [65, 90].

In all cases, algorithms admit fast linear convergence rates. The majority of approaches assumes a *first-order* oracle: information of f is provided through its gradient $\nabla f(X)$. For MS, $\nabla f(X) = -2\mathcal{A}^*(y - \mathcal{A}(X))$, which requires $O(\mathrm{T}_{\mathrm{map}})$ complexity, where $\mathrm{T}_{\mathrm{map}}$ denotes the time required to apply linear map (or its adjoint $\mathcal{A}^*$) $\mathcal{A}$. Formulations (3)–(4) require at least one top-r SVD calculation per iteration; this translates into additional $O(mnr)$ complexity.

Problem (3) can be factorized as follows:

$$\underset{U\in\mathbb{R}^{n\times r}, V\in\mathbb{R}^{p\times r}}{\text{minimize}} \quad f(UV^\top) := \tfrac{1}{2}\cdot\|y - \mathcal{A}(UV^\top)\|_2^2. \tag{5}$$

For this case, the gradient of f with respect to U and V can be computed respectively as $\nabla_U f(UV^\top) := \nabla f(X)V$ and $\nabla_V f(UV^\top) := \nabla f(X)^\top U$. This translates into $2\cdot O(\mathrm{T}_{\mathrm{map}}+mnr)$ time complexity. This excludes performing any SVD calculations per iteration. Thus, if there exist linearly convergent algorithms for (5), intuition indicates that we could obtain lower computational complexity.

1.1.2. Logistic PCA and low-rank estimation on binary data. Finding low-rank approximations of binary matrices has gained a lot of interest recently, due to the wide appearance of categorical responses in real-world applications [87, 31, 56, 94, 44, 74]. The authors of [91, 33] propose logistic PCA, where each binary data vector is assumed to follow the multivariate Bernoulli distribution, parametrized by the principal components that live in an r-dimensional subspace.

To formulate the problem, let $Y \in \{0,1\}^{m\times n}$ be the observed binary matrix, where each of the m rows stores an n-dimensional binary feature vector. Further, assume that each entry Y_{ij} is drawn from a Bernoulli distribution with mean q_{ij}, according to $\mathbb{P}[Y_{ij} \mid q_{ij}] = q_{ij}^{Y_{ij}}\cdot(1-q_{ij})^{1-Y_{ij}}$. Define the log-odds parameter $X_{ij} = \log(\frac{q_{ij}}{1-q_{ij}})$ and the logistic function $\sigma(X_{ij}) = \left(1+e^{-X_{ij}}\right)^{-1}$. Then, we equivalently have $\mathbb{P}[Y_{ij} \mid X_{ij}] = \sigma(X_{ij})^{Y_{ij}}\cdot\sigma(-X_{ij})^{1-Y_{ij}}$, or in matrix form, $\mathbb{P}[Y \mid X] = \prod_{ij}\sigma(X_{ij})^{Y_{ij}}\cdot\sigma(-X_{ij})^{1-Y_{ij}}$, where we assume independence among entries of Y. The negative log-likelihood for log-odds parameter X is given by

$$f(X) := -\sum_{ij} \left(Y_{ij} \cdot \log \sigma(X_{ij}) + (1 - Y_{ij}) \cdot \log \sigma(-X_{ij})\right).$$

Assuming a compact, i.e., low-rank, representation for the latent variable X, we end up with the following optimization problem:

$$\begin{aligned} &\underset{X \in \mathbb{R}^{m \times n}}{\text{minimize}} \quad f(X) := -\sum_{ij} \left(Y_{ij} \cdot \log \sigma(X_{ij}) + (1 - Y_{ij}) \cdot \log \sigma(-X_{ij})\right) \\ &\text{subject to} \quad \operatorname{rank}(X) \leq r. \end{aligned} \tag{6}$$

As we see later in the text, the objective criterion is just a smooth convex loss function.

State-of-the-art approaches. In [31], the authors consider the problem of *sign prediction* of edges in a signed network and cast it as a low-rank matrix completion problem: The proposed algorithmic solution follows (stochastic) gradient descent motions; however, no guarantees are provided. Johnson [56] utilizes logistic PCA for collaborative filtering on implicit feedback data (page clicks and views, purchases, etc.): to find a local minimum, an alternating gradient descent procedure is used, with no guarantees. A similar alternating gradient descent approach is followed in [87], with no known theoretical guarantees.

Parameterization by the latent factors U, V leads to the following optimization criterion:

$$\underset{U \in \mathbb{R}^{m \times r},\ V \in \mathbb{R}^{n \times r}}{\text{minimize}} \quad f(UV^\top) := -\sum_{ij} \left(Y_{ij} \cdot \log \sigma(U_i V_j^\top) + (1 - Y_{ij}) \cdot \log \sigma(-U_i V_j^\top)\right), \tag{7}$$

where U_i, V_j represent the ith and jth rows of U and V, respectively.

2. Preliminaries. For matrices $X, Y \in \mathbb{R}^{m \times n}$, $\langle X, Y \rangle = \operatorname{Tr}\left(X^\top Y\right)$ represents their inner product. We use $\|X\|_F$ and $\sigma_1(X)$ for the Frobenius and spectral norms of a matrix, respectively; also $\|X\|_2$ denotes the spectral norm. $\sigma_i(X)$ is the ith singular value of X. For a rank-r matrix $X = UV^\top$, the gradient of f w.r.t. U and V is $\nabla f(UV^\top)V$ and $\nabla f(UV^\top)^\top U$, respectively. We will also use the terms $\nabla_U f(UV^\top) := \nabla f(UV^\top)V$ and $\nabla_V f(UV^\top) := \nabla f(UV^\top)^\top U$.

Given a matrix X, we denote its best rank-r approximation with X_r. We denote the optimum point as $X_r^\star$, both (i) in the case where we intentionally restrict our search to obtain a rank-r approximation of $X^\star$—while $\operatorname{rank}(X^\star) > r$—and (ii) in the case where $X^\star \equiv X_r^\star$, i.e., by default, the optimum point is of rank r.

An important issue in optimizing f over the factored space is the existence of nonunique possible factorizations for a given X. Our analysis requires a notion of distance to the low-rank solution $X_r^\star$ over the factors. Among infinitely many possible decompositions of $X_r^\star$, we focus on the set of "equally footed" factorizations [92, 50]:

$$\begin{aligned} \mathcal{X}_r^\star = \Big\{ (U^\star, V^\star) :\ & U^\star \in \mathbb{R}^{m \times r}, V^\star \in \mathbb{R}^{n \times r}, \\ & U^\star V^{\star\top} = X_r^\star, \sigma_i(U^\star) = \sigma_i(V^\star) = \sigma_i(X_r^\star)^{1/2}\ \forall i \in [r] \Big\}. \end{aligned} \tag{8}$$

Note that $(U^\star, V^\star) \in \mathcal{X}_r^\star$ if and only if $U^\star = A^\star \Sigma^{\star 1/2} R$, $V^\star = B^\star \Sigma^{\star 1/2} R$, where $A^\star \Sigma^\star B^\star$ is the SVD of $X_r^\star$, and $R \in \mathbb{R}^{r \times r}$ is an orthogonal matrix.

Given a pair (U, V), we define the distance to $X_r^\star$ as

$$\textsc{Dist}\,(U, V; X_r^\star) = \min_{(U^\star, V^\star) \in \mathcal{X}_r^\star} \left\| \begin{bmatrix} U \\ V \end{bmatrix} - \begin{bmatrix} U^\star \\ V^\star \end{bmatrix} \right\|_F.$$

Assumptions. We consider applications that can be described either (i) by (restricted) *strongly* convex functions f with *gradient Lipschitz continuity* or (ii) by convex functions f that have only (restricted) Lipschitz continuous gradients. We state these standard definitions below.

Definition 2.1. *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a convex differentiable function. Then, f is gradient Lipschitz continuous with parameter L (or L-smooth) if*

$$\|\nabla f(X) - \nabla f(Y)\|_F \le L \cdot \|X - Y\|_F \ \ \forall X, Y \in \mathbb{R}^{m\times n}. \tag{9}$$

The function is restricted smooth if the above holds only for all rank-r X, Y.

Definition 2.2. *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be convex and differentiable. Then, f is μ-strongly convex if*

$$f(Y) \ge f(X) + \langle \nabla f(X), Y - X\rangle + \tfrac{\mu}{2}\|Y - X\|_F^2 \ \ \forall X, Y \in \mathbb{R}^{m\times n}. \tag{10}$$

The function is restricted strongly convex if the above holds only for all rank-r X, Y.

The factored gradient descent algorithm. Part of our contribution is inspired by [14], where the factored gradient descent (FGD) algorithm is proposed. For completeness, we describe here the problem they consider and the proposed algorithm. They [14] consider the problem

$$\underset{X\in\mathbb{R}^{n\times n}}{\text{minimize}} \quad f(X) \quad \text{subject to} \quad X \succeq 0$$

and propose the following first-order recursion for its solution:

$$U_{t+1} = U_t - \eta \cdot \nabla f(U_t U_t^\top) \cdot U_t.$$

A key property in their analysis is the positive semidefiniteness of the feasible space. For a proper initialization and step size, [14] shows sublinear and linear convergence rates toward optimum, depending on the nature of f.

3. The BFGD algorithm.

We provide an overview of the BFGD algorithm for two problem settings in (1): (i) f being an L-smooth convex function and (ii) f being L-smooth and μ-strongly convex; the results for restricted assumptions naturally generalize. For both cases, we assume a good initialization point $X_0 = U_0 V_0^\top$; see section 5.

BFGD is built upon nonconvex gradient descent over U and V, written as

$$U_{t+1} = U_t - \eta \cdot \nabla_U f(U_t V_t^\top), \quad V_{t+1} = V_t - \eta \cdot \nabla_V f(U_t V_t^\top). \tag{11}$$

When f is convex and smooth, BFGD follows exactly the motions in (11); in the case where f is also strongly convex, BFGD is based on a different set of recursions, which we discuss in more detail later in the text.

3.1. Reduction to FGD: When f is convex and L-smooth. In [51], the authors describe a simple technique to transform (1) into problems where we look for a square and PSD solution. The key idea is to *lift* the problem and introduce a stacked matrix of the two factors:

$$W = \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{(m+n)\times r}.$$

Then, we optimize over a new function $\hat{f} : \mathbb{R}^{(m+n)\times(m+n)} \to \mathbb{R}$ defined as

$$\hat{f}\left(WW^\top\right) = \hat{f}\left(\begin{bmatrix} UU^\top & UV^\top \\ VU^\top & VV^\top \end{bmatrix}\right) = f(UV^\top).$$

Following this idea, we utilize algorithms designed only to work on square and PSD-based instances, where f is just L-smooth. Here, we use the FGD algorithm of [14] on the W-space, as follows:

$$W_{t+1} = W_t - \eta \cdot \nabla_W \hat{f}(W_t W_t^\top). \tag{12}$$

It is easy to verify the following remark.

Remark 1. Define $\hat{f}([\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}]) = \frac{1}{2}f(B) + \frac{1}{2}f(C^\top)$ for $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{n\times m}$, $D \in \mathbb{R}^{n\times n}$. Then FGD for minimizing $\hat{f}(WW^\top)$ with the stacked matrix $W = [U^\top, V^\top]^\top \in \mathbb{R}^{(m+n)\times r}$ is equivalent to (11).

A natural question is whether this reduction gives a desirable convergence behavior. To answer this, we need to characterize the properties of $\hat{f}$.

Proposition 3.1. *If f is convex and L-smooth, then $\hat{f}$ is convex and $\frac{L}{2}$-smooth.*

Proof. For any $Z_1 = [\begin{smallmatrix} A_1 & B_1 \\ C_1 & D_1 \end{smallmatrix}], Z_2 = [\begin{smallmatrix} A_2 & B_2 \\ C_2 & D_2 \end{smallmatrix}] \in \mathbb{R}^{(m+n)\times(m+n)}$, we have

$$\begin{aligned} \left\|\nabla\hat{f}(Z_1) - \nabla\hat{f}(Z_2)\right\|_F &= \tfrac{1}{2} \cdot \sqrt{\|\nabla f(B_1) - \nabla f(B_2)\|_F^2 + \left\|\nabla f(C_1^\top) - \nabla f(C_2^\top)\right\|_F^2} \\ &\le \tfrac{L}{2} \cdot \sqrt{\|B_1 - B_2\|_F^2 + \|C_1 - C_2\|_F^2} \\ &\le \tfrac{L}{2} \cdot \|Z_1 - Z_2\|_F, \end{aligned}$$

where the first inequality follows from the L-smoothness of f. ∎

Based on the above proposition, we use FGD to solve (2) with $\hat{f}$: its procedure is exactly (11), with a different step size than [14], due to tighter analysis:

$$\eta \le \frac{1}{20L\left\|\begin{bmatrix} U_0 \\ V_0 \end{bmatrix}\right\|_2^2 + 3\|\nabla f(U_0 V_0^\top)\|_2}. \tag{13}$$

While one can rely on the sublinear convergence analysis from [14], we provide a new guarantee with a weaker initial condition; see section 4.

3.2. Using BFGD when f is L-smooth and strongly convex. Assume f satisfies both properties in Definitions 2.1 and 2.2. In this case, we cannot simply rely on the lifting technique as above since $\hat{f}$ is clearly not strongly convex.[3] Here, we consider a slight variation, where we appropriately regularize the objective and force the solution pair $(\widehat{U}, \widehat{V})$ to be "balanced." This regularization is based on the set of optimal pairs $(U^\star, V^\star)$ in $\mathcal{X}_r^\star$, as defined in (8). Given $\mathcal{X}_r^\star$, the equivalent optimization problem that "forces" convergence to balanced $(U^\star, V^\star)$ is

$$\underset{U\in\mathbb{R}^{m\times r},\ V\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad f(UV^\top) + \lambda \cdot g(U^\top U - V^\top V), \tag{14}$$

where $g : \mathbb{R}^{r\times r} \to \mathbb{R}$ is an additional convex *regularizer*. We require the following:

- g is convex and minimized at zero point, i.e., $\nabla g(0) = 0$.
- The gradient, $\nabla g(U^\top U - V^\top V) \in \mathbb{R}^{r\times r}$, is symmetric for any such pair.
- g is μ_g-strongly convex and L_g-smooth.

Under such assumptions, the addition of g in the objective just restricts the set of optimum points to be "balanced", *i.e., the minimizer of* (14) *minimizes also* (2).[4]

The necessity of the regularizer. The theoretical guarantees of BFGD heavily depend on the condition number of the pair $(U^\star, V^\star)$ the algorithm converges to. In particular, one of the requirements of BFGD is that every estimate U_t (resp., V_t) be "relatively close" to the convergent point $U^\star$ (resp., $V^\star$), such that their distance $\|U_t - U^\star\|_F$ is bounded by a function of $\sigma_r(U^\star)$, for all t. Though, for arbitrarily ill-conditioned $(U^\star, V^\star) \notin \mathcal{X}_r^\star$, such a condition might not be easily satisfied by BFGD per iteration[5], unless we "force" the sequence of estimates (U_t, V_t) for all t to converge to a better conditioned pair $(U^\star, V^\star)$. This is the key role of regularizer g: it guarantees putative estimates U_t and V_t are not too ill-conditioned, per iteration.

An example of g is the Frobenius norm (weighted by $\mu/2$), as proposed in [92]. Other examples are sums of elementwise (at least) μ_g-strongly convex and (at most) L_g-gradient Lipschitz functions (of the form $g(X) = \sum_{i,j} g_{ij}(X_{ij})$) with the optimum at zero. Any such regularizer results provably in convergence; see section 7.2 for a toy example where the addition of g leads to faster convergence rate in practice.

The BFGD algorithm. BFGD is a first-order, gradient descent algorithm for (14) that operates on the factored space (U, V) in an alternating fashion. Principal components of BFGD is a proper step size selection and a "decent" initialization point. BFGD can be considered as the nonsquared extension of the FGD algorithm in [14], which is specifically designed to solve problems as in (2), for $U = V$ and $m = n$. The key differences with FGD, though, other than the necessity of a regularizer g, are as follows:

- Our analysis leads to provable convergence results in the nonsquare case. Such a result cannot be trivially obtained from [14].

[3] To see this, observe that $\hat{f}$ selects only the off-block diagonal elements and neglects $UU^\top$ and $VV^\top$. Computing the Hessian of $\hat{f}$ function, it is easy to observe that it has zero eigenvalues (due to the elements on the diagonal that are not selected). Thus, $\hat{f}$ is not strongly convex.

[4] In particular, for any rank-r solution $UV^\top$ in (2), there is a factorization $(\tilde{U}, \tilde{V})$ minimizing g with the same function value $f(\tilde{U}\tilde{V}^\top) = f(UV^\top)$, which are $\tilde{U} = A\Sigma^{\frac{1}{2}}$, $\tilde{V} = B\Sigma^{\frac{1}{2}}$, where $UV^\top = A\Sigma B^\top$ is the SVD.

[5] Even if $UV^\top$ is close to $U^\star V^{\star\top}$, the condition numbers of U, V can be larger than that of $UV^\top$.

Algorithm 1. BFGD for smooth and strongly convex f.

Input: Function f, target rank r, # iterations T.
Set initial values for U_0, V_0
Set step size η as in (15).
for $t = 0$ to $T-1$ **do**
  $U_{t+1} = U_t - \eta\left(\nabla_U f(U_t V_t^\top) - \lambda \cdot \nabla_U g(U_t^\top U_t - V_t^\top V_t)\right)$
  $V_{t+1} = V_t - \eta\left(\nabla_V f(U_t V_t^\top) - \lambda \cdot \nabla_V g(U_t^\top U_t - V_t^\top V_t)\right)$
end for
Output: $X = U_T V_T^\top$.

- The main recursion is different in the two schemes: in the nonsquared case, we update the left and right factors $(U,\ V)$ with a different rule, according to which

$$U_{t+1} = U_t - \eta\left(\nabla_U f(U_t V_t^\top) + \lambda \cdot \nabla_U g(U_t^\top U_t - V_t^\top V_t)\right),$$
$$V_{t+1} = V_t - \eta\left(\nabla_V f(U_t V_t^\top) + \lambda \cdot \nabla_V g(U_t^\top U_t - V_t^\top V_t)\right).$$

  The parameter $\lambda > 0$ is arbitrarily chosen.
- Due to this new rule, a slightly different and proper step size selection is required for BFGD. Our step size is selected as follows:

$$\eta \le \frac{1}{12 \cdot \max\{L,\ L_g\} \cdot \left\| \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} \right\|_2^2}. \tag{15}$$

  Compared to the step size proposed in [14] (which is of the same form with (13)), our analysis drops the dependence to $\|\nabla f(\cdot)\|_2$ at the denominator. This leads to a faster computed η and highlights the nonnecessity of this term for proof of convergence, i.e., the $\|\nabla f(\cdot)\|_2$ term is sufficient but not necessary.

The scheme is described in Algorithm 1. As we show next, constant η (15) is sufficient to lead to attractive convergence rates for BFGD, for f L-smooth and μ-strongly convex.

4. Local convergence for BFGD. To provide local convergence results, we assume that there is a known "good" initialization which ensures the following.

Assumption 1. Define $\kappa = \frac{\max\{L, L_g\}}{\min\{\mu, \mu_g\}}$, where μ_g and L_g are the strong convexity and smoothness parameters of g, respectively. Then, we assume we are provided with a "good" initialization point $X_0 = U_0 V_0^\top$ such that

$$\textsc{Dist}(U_0, V_0; X_r^\star) \le \frac{\sqrt{2}\cdot\sigma_r(X_r^\star)^{1/2}}{10\sqrt{\kappa}} \qquad \text{(strongly convex and smooth } f\text{)}.$$

For the case where f is just smooth, we assume

$$\textsc{Dist}(U_0, V_0; X_r^\star) \le \frac{\sqrt{2}\cdot\sigma_r(X_r^\star)^{1/2}}{10} \qquad \text{(smooth f)}.$$

For our analysis, we will use the following step size assumptions:

$$\widehat{\eta} \le \frac{1}{8\max\{L, L_g\} \cdot \left\| \begin{bmatrix} U_t \\ V_t \end{bmatrix} \right\|_2^2} \qquad \text{(strongly convex and smooth } f\text{)}, \tag{16}$$

$$\widehat{\eta} \le \frac{1}{15L \left\| \begin{bmatrix} U_t \\ V_t \end{bmatrix} \right\|_2^2 + 3\|\nabla f(U_t V_t^\top)\|_2} \qquad \text{(smooth } f\text{)}. \tag{17}$$

While these step sizes are different from the ones we use in practice, there is a constant-fraction connection between $\widehat{\eta}$ and η.

Lemma 4.1. *Let (U_0, V_0) be such that Assumption 1 is satisfied. Then, (16) holds if (15) is satisfied, and (17) holds if (13) is satisfied.*

The proof is provided in Appendix A. By this lemma, our analysis below is equivalent—up to constants—to that if we were using the original step size η of the algorithm. However, for clarity reasons and ease of exposition, we use $\widehat{\eta}$ below.

For the case of strongly convex f, both Assumption 1 and the step size depend on the strong convexity and smoothness parameters of g. When μ and L are known a priori, this dependency can be removed since one can choose g such that at least μ-restricted strongly convex and at most L-smooth. Then, κ becomes the condition number of f, and the step size depends only on L.

4.1. Linear local convergence rate for f L-smooth and μ-strongly convex. The following theorem proves that, under proper initialization, `BFGD` admits a linear convergence rate, when f is both L-smooth and μ-restricted strongly convex.

Theorem 4.2. *Suppose that f is L-smooth and μ-strongly convex, and the regularizer g is L_g-smooth and μ_g-strongly convex. Define $\mu_{\min} := \min\{\mu, \mu_g\}$ and $L_{\max} := \max\{L, L_g\}$. Denote the unique minimizer of f as $X^\star \in \mathbb{R}^{m\times n}$ and assume that $X^\star$ is of arbitrary rank. Let $\widehat{\eta}$ be defined as in (16). If the initial point (U_0, V_0) satisfies Assumption 1, then the `BFGD` algorithm in Algorithm 1 converges linearly to $X_r^\star$, within error $O(\sqrt{\frac{\kappa}{\sigma_r(X_r^\star)}}\,\|X^\star - X_r^\star\|_F)$, according to the recursion*

$$\textsc{Dist}(U_{t+1}, V_{t+1}; X_r^\star)^2 \le \gamma_t \cdot \textsc{Dist}(U_t, V_t; X_r^\star)^2 + \widehat{\eta} L \,\|X^\star - X_r^\star\|_F^2 \tag{18}$$

for every $t \ge 0$, where the contraction parameter γ_t satisfies

$$\gamma_t = 1 - \widehat{\eta} \cdot \tfrac{\mu_{\min}\cdot\sigma_r(X_r^\star)}{5} \ge 1 - \tfrac{\mu_{\min}}{65\cdot L_{\max}} \cdot \tfrac{\sigma_r(X_r^\star)}{\sigma_1(X_r^\star)} > 0.$$

The proof is provided in Appendix B. The theorem states that if $X^\star$ is (nearly) low rank, the iterates converge to a close neighborhood of $X_r^\star$. The above result can also be expressed w.r.t. the function value $f(UV^\top)$, as follows.

Corollary 4.3. *Under the same initial condition with Theorem 4.2, Algorithm 1 satisfies the following recursion w.r.t. the distance of function values:*

$$f(U_t V_t^\top) - f(X^\star) \le \gamma^t \cdot \sigma_1(X^\star) \cdot \Big(f(U_0 V_0^\top) - f(X^\star)\Big) + \frac{\sqrt{\mu L}}{\sigma_r(X^\star)}\,\|X^\star - X_r^\star\|_F^2\,.$$

4.2. Local sublinear convergence. In section 3.1, we showed that a lifting technique can reduce our problem (2) to a rank-constrained semidefinite program, and applying FGD from [14] is exactly BFGD (11). While the sublinear convergence guarantee of FGD can also be applied to our problem, we provide an improved result.

Theorem 4.4. *Suppose that f is L-smooth with a minimizer $X^\star \in \mathbb{R}^{m\times n}$. Let $\hat{X}_r$ be any target rank-r matrix, and let $\widehat{\eta}$ be defined as in (17). If the initial point $X_0 = U_0V_0^\top$, $U_0 \in \mathbb{R}^{m\times r}$ and $V_0 \in \mathbb{R}^{n\times r}$, satisfies Assumption 1, then FGD converges with rate $O(1/T)$ to a tolerance value according to*

$$f(U_TV_T^\top) - f(U^\star V^{\star\top}) = \hat{f}(W_TW_T^\top) - \hat{f}(W^\star W^{\star\top}) \le \frac{10\cdot \textsc{Dist}(U_0,V_0;X_r^\star)^2}{\eta T}.$$

Theorem 4.4 guarantees a local sublinear convergence with a looser initial condition. While [14] requires $\min_{R\in O(r)} \|W - W^\star R\|_F \le \frac{\sigma_r^2(W^\star)}{100\sigma_1^2(W^\star)}\cdot\sigma_r(W^\star)$, our result requires that the initial distance to the $W^\star$ is merely a constant factor of $\sigma_r(W^\star)$.

5. Initialization. Our main theorem guarantees linear convergence in the factored space given that the initial point (U_0, V_0) is within a ball around the closest target factors, with radius $O(\kappa^{-1/2}\sigma_r(X_r^\star)^{1/2})$. To find such a solution, we propose an extension of the initialization in [14].

Lemma 5.1. *Consider $U_0V_0^\top$ which is the best rank-r approximation of*

$$X_0 = -\tfrac{1}{L}\nabla f(0). \tag{19}$$

Then we have $\left\|U_0V_0^\top - X_r^\star\right\|_F \le 2\sqrt{2(1-\frac{1}{\kappa})}\,\|X^\star\|_F + 2\,\|X^\star - X_r^\star\|_F$.

The proof can be found in Appendix D. Combined with Lemma 5.14 in [92], which transforms a good initial solution from the original space to the factored space, the following corollary gives one sufficient condition for global convergence of BFGD with the SVD of (19) as initialization.

Corollary 5.2. *If $\|X^\star - X_r^\star\|_F \le \frac{\sigma_r(X^\star)}{100\sqrt{\kappa}}$, $\kappa \le 1 + \frac{\sigma_r(X_r^\star)^2}{4608\cdot\|X_r^\star\|_F^2}$, then the initial solution $U_0 = A_0\Sigma_0^{1/2}$, $V_0 = B_0\Sigma_0^{1/2}$, where $A_0\Sigma_0B_0$ is the SVD of $-\frac{1}{L}\nabla f(0)$ satisfies the initial condition of Theorem 4.2.*

The proof is easily derived by substituting its assumptions in Lemma 5.1. Corollary 5.2 requires weaker conditions than [14] in order for Theorem 4.2 to be transformed to *global guarantees*. While our theoretical results can only guarantee global convergence for a well-conditioned problem (κ close to one), we show in the experiments that the algorithm performs well in practice, even when conditions are unverifiable a priori.

6. Related work. This is not the first time such transformations have been considered in practice. Burer and Monteiro [21, 22] popularized these ideas for solving SDPs: their approach embeds the PSD and linear constraints into the objective and applies low-rank variable reparameterization. While the constraint considered here is of a different nature—i.e., rank constraint versus PSD constraint—the motivation is similar. See also [19] for a more recent result on SDPs.

Table 1

Summary of selected nonconvex *solvers for low-rank inference problems. X, $UU^\top$, and $UV^\top$ denote the setting the algorithm works at; X means that no factorization is considered. "Rate" describes the convergence rate; "(Sub)linear" denotes that both sublinear and linear rates are proved, depending on the nature of f. "Function f" denotes the problem cases covered by each algorithm: "MS" and "MC" stand for matrix sensing and completion, respectively; "Convex f" corresponds to standard smooth and strongly convex functions f. For the case of [30], "Generic $f^\dagger$" corresponds to a specific class of functions that can be even concave but should satisfy specific conditions; see [30].*

Algorithm	X	$UU^\top$	$UV^\top$	Rate	Function f
[54]	✓	✗	✗	Linear	MS
[90]	✓	✗	✗	Linear	MS
[65]	✓	✗	✗	Linear	MS
[49]	✗	✓	✗	Sublinear	MS
[68]	✗	✓	✗	Sublinear	Convex f
[30]	✗	✓	✗	(Sub)linear	Generic $f^\dagger$
[14]	✗	✓	✗	(Sub)linear	Convex f
[51]	✗	✓	✓	Sublinear	Convex f
[92]	✗	✓	✓	Linear	MS
[107]	✗	✓	✓	Linear	MS
[48]	✗	✓	✓	Linear	MC
[54]	✗	✓	✓	Linear	MC, MS
[89]	✗	✓	✓	Linear	MC
[106]	✗	✓	✓	Linear	Strongly convex f
This work	✗	✓	✓	(Sub)linear	Convex f

We provide an overview of algorithms that solve instances of (2). For discussions on methods that operate on X directly, we refer the reader to [2, 51, 65] for more details; see also Table 1 for an overview of the discussion below. We divide our discussion into two problem settings: (i) $X^\star$ is square and PSD and (ii) $X^\star$ is nonsquare.

Square and PSD $X^\star$. A rank-r $X \in \mathbb{R}^{n\times n}$ is PSD if and only if it can be factored as $X = UU^\top$ for $U \in \mathbb{R}^{n\times r}$. This is a special case of our problem, where $m = n$ and (1) includes a PSD constraint. Thus, (2) takes the form

$$\underset{U\in\mathbb{R}^{n\times r}}{\text{minimize}} \quad f(UU^\top), \qquad \text{where } r = \operatorname{rank}(X^\star) \le n. \tag{20}$$

Several recent works have studied (20). For the special case where f is a least-squares objective for an underlying linear system, [92] and [107] propose gradient descent schemes that function on the factor U. Both studies employ careful initialization (performing few iterations of `SVP` [53] for the former and using a spectral initialization procedure for the latter) and step size selection, in order to prove convergence.[6] However, their analysis is designed only for least-squares instances of f. Further discussion on their step size selection/initialization is provided in section 7.

[6] Recently, [42] and [15] proved that $UU^\top$ factorization introduces no spurious local minima for the cases of matrix completion and sensing, respectively: random initialization eventually leads to convergence to the optimal $X^\star$ (or close to $X^\star$ in the nearly low-rank case).

The work of [30] proposes a first-order algorithm for (20), where f is more generic. The algorithmic solution proposed can handle additional constraints on the factors U; the nature of these constraints depends on the problem at hand.[7] For each problem, a set of assumptions needs to be satisfied, i.e., faithfulness, local descent, local Lipschitz, and local smoothness conditions. Under such assumptions and with proper initialization, one can prove convergence with an $O(1/\varepsilon)$ or $O(\log(1/\varepsilon))$ rate, depending on the nature of f, and for problems that even fail to be locally convex. Our work provides similar convergence results but for a set of assumptions used more in practice.

Bhojanapalli, Kyrillidis, and Sanghavi [14] propose the `FGD` algorithm for (20). `FGD` is also a first-order scheme; a key ingredient for convergence is a novel step size selection that can be used for any f, as long as it is (restricted) gradient Lipschitz continuous; when f is further (restricted) strongly convex, their analysis leads to faster convergence rates. An extension of these ideas to some constrained cases can be found in [83].

Nonsquare $X^\star$. Jain, Netrapalli, and Sanghavi [54] propose `AltMinSense`, an alternating minimization algorithm for matrix sensing and matrix completion problems. This is one of the first works to prove linear convergence in solving (2) for the MS model. Hardt and Wooders [48] improve upon [54] for the case of reasonably well-conditioned matrices. Their algorithm handles problem cases with bad condition number and gaps in their spectrum [102]. Recently, [92] extended the `Procrustes Flow` algorithm to the nonsquare case, where gradient descent, instead of exact alternating minimization, is utilized. Zheng and Lafferty [108] extended the first-order method of [30] for matrix completion to the rectangular case. All the studies above focus on the case of least-squares objective f.

Sun and Luo [89] generalize the results in [54, 48]: the authors show that, under common incoherence conditions and sampling assumptions, most first-order variants indeed converge to the low-rank ground truth $X^\star$. Both the theory and the algorithm proposed are restricted to the matrix completion objective.

Recently, [106]—based on the inexact first-order oracle, previously used in [7]—proved that linear convergence is guaranteed if $f(UV^\top)$ is strongly convex over either U and V, when the other is fixed. While the technique applies for generic f and for nonsquare X, the authors provide algorithmic solutions only for matrix completion/matrix sensing settings.[8] Their algorithm requires QR-decompositions after each U, V update; this is required in order to control the notion of inexact first-order oracle.

Finally, we mention relevant optimization methods that operate over manifolds and admit tailored solvers [38]; see [60, 17, 20, 105, 93] and references therein for applications in matrix completion. Among the most established work on the field, [57] presents a second-order method for (1), based on manifold optimization over the set of an orthonormal equivalence class of matrices. The proposed algorithm can accommodate constraints and enjoys monotonic decrease of the objective function (in contrast to [21, 22]), featuring quadratic local convergence. In practice, the per iteration complexity is dominated by the extraction of the

[7] Any additional constraints should satisfy the *faithfulness* property: a constraint set $\mathcal{C}$ is faithful if for each $U \in \mathcal{C}$, within some bounded radius from optimal point, we are guaranteed that the closest (in the Euclidean sense) rotation of optimal $U^\star$ lies within $\mathcal{U}$.

[8] For example, in the gradient descent case, the step size proposed depends on RIP [85] constants and it is not clear what a good step size would be in other problem settings.

eigenvector, corresponding to the smallest eigenvalue, of a $n \times n$ matrix—and only when the current estimate of rank satisfies some conditions. See also [62].

Mishra et al. [79] focus on low-rank matrix approximations and propose a gradient descent algorithm that resembles ours (using manifold notation and techniques). In contrast to [79], our work provides theoretical guarantees on the performance of such first-order algorithms. See also [80, 78]. Kressner, Steinlechner, and Vandereycken [64] focus on tensor matrix completion.[9] As above, that work focuses on describing the notation and operations performed by manifold algorithms, without presenting any convergence (or convergence rates) results of the algorithm. Finally, [109] provides convergence results for a first-order manifold scheme with an *asymptotic* flavor, i.e., assumes the number of iterates converges to infinity. Contrarily, we provide nonasymptotic results of simple gradient descent to the optimal point, after proper initialization. Zhou et al. [109] consider a problem criterion similar to ours but under different assumptions: in our case, the assumptions appear in many signal processing/machine learning tasks (smoothness, strong convexity), while [109] assumes that f admits a differential extension f_F on a neighborhood of a manifold space.

7. Experiments.

7.1. The complexity of SVD and matrix-matrix multiplication.

To provide an idea of how matrix-matrix multiplication scales, in comparison with truncated SVD,[10] we compare it with some state-of-the-art SVD subroutines: (i) the MATLAB `svds` subroutine, based on the `ARPACK` software package [70], (ii) a collection of implicitly restarted Lanczos methods for fast truncated SVD and symmetric eigenvalue decompositions (`irlba`, `irlbablk`, `irblsvds`) [6],[11] (iii) the limited memory block Krylov subspace optimization for computing dominant SVDs (`LMSVD`) [73], and (iv) the `PROPACK` software package [67]. We consider random realizations of matrices in $\mathbb{R}^{m\times n}$ (without loss of generality, assume $m = n$) for varying values of m. For SVD computations, we look for the best rank-r approximation for varying values of r. In the case of matrix-matrix multiplication, we record the time required for the computation of two matrix-matrix multiplications of matrices $\mathbb{R}^{m\times m}$ and $\mathbb{R}^{m\times r}$, which is equivalent to the computational complexity required in our scheme.

Figure 1, left panel, shows execution time results for the algorithms under comparison, as a function of the dimension m. Rank r is fixed to $r = 100$. While both SVD and matrix multiplication procedures are known to have $O(m^2 r)$ complexity, it is obvious that the latter on dense matrices is at least two orders of magnitude faster than the former. In Table 2, we also report the approximation guarantees of some faster SVD subroutines, as compared to `svds`: while `irblablk` seems to be faster, it returns a very rough approximation of the singular values, when r is relatively large. Similar findings are depicted in Figure 1, middle and right panels.

[9] From a theoretical standpoint, matrix completion is outside the scope of this paper, as it does not satisfy strong convexity; here, we focus on problems that satisfy (restricted) smoothness and (restricted) strong convexity, while matrix completion problems require other regulatory conditions such as incoherence.

[10] Here, we consider algorithmic solutions where both SVD and matrix-matrix multiplication computations are performed with high accuracy. One might consider *approximate* SVD—see the excellent monograph [47]—and matrix-matrix multiplication approximations—see [36, 37, 66, 32]; we believe that studying such alternatives is an interesting direction to follow for future work.

[11] IRLBA stands for implicitly restarted Lanczos bidiagonalization algorithms.

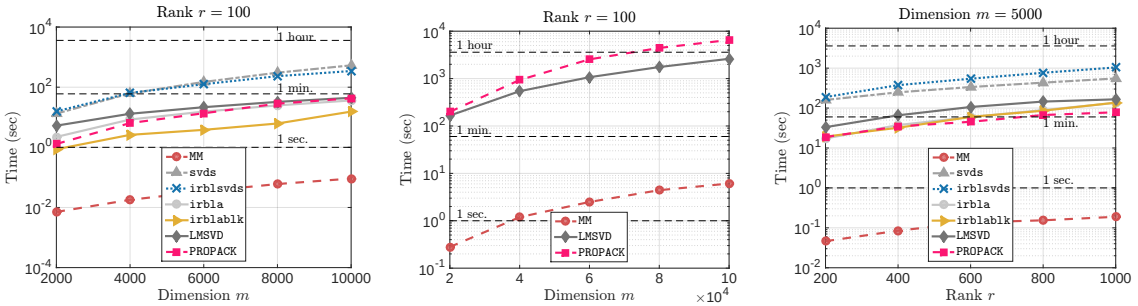

Figure 1. *Comparison of SVD procedures versus matrix-matrix (MM) multiplication. Left panel: Varying dimension m and constant rank $r = 100$. Middle panel: Similar to left panel where m scales larger and we focus on a subset of SVD algorithms that can scale up. Right panel: Varying rank values and constant dimension $m = 5 \cdot 10^3$.*

Table 2

Approximation errors of singular values, in the form $\frac{\|\widehat{\Sigma}-\Sigma^\star\|_F}{\|\Sigma^\star\|_F}$. Here, $\widehat{\Sigma}$ denotes the diagonal matrix, returned by SVD subroutines, containing r top singular values; we use `svds` *to compute the reference matrix $\Sigma^\star$ that contains top-r singular values of the input matrix. Observe that some algorithms deviate significantly from the "ground-truth": this is due to either early stopping (only a subset of singular values could be computed) or accumulating approximation error.*

Algorithm	Error $\frac{\|\widehat{\Sigma}-\Sigma^\star\|_F}{\|\Sigma^\star\|_F}$, where $\Sigma^\star$ is diagonal matrix with top r singular values from `svds`				
	$m = 2 \cdot 10^3$	$m = 4 \cdot 10^3$	$m = 6 \cdot 10^3$	$m = 8 \cdot 10^3$	$m = 10^4$
`irblsvds`	3.63e-15	4.33e-09	8.11e-11	4.79e-12	5.82e-10
`irbla`	6.00e-15	9.01e-07	1.05e-04	2.99e-04	7.29e-04
`irblablk`	1.48e+03	1.67e+03	1.24e+03	1.45e+03	7.91e+11
`LMSVD`	2.14e-14	4.49e-12	3.94e-11	1.33e-10	7.30e-10
`PROPACK`	4.10e-12	2.46e-10	1.63e-12	7.90e-12	3.55e-11

7.2. The role of regularizer g. As discussed in section 3.2, g forces our algorithm to converge to a well-conditioned factorization of $X^\star$. This regularizer not only enables us to control and guarantee convergence of `BFGD` but also provides a better convergence rate, as we know next.

Figure 2, left panel, shows the convergence behavior of `BFGD`, when f and $f + g$ are used, with an ill-conditioned initial point (U_0, V_0). It is obvious from the convergence plot that adding the regularizer results in faster convergence to an optimum. This difference in convergence rate is due to dependency on the condition numbers of $U^\star$ and $V^\star$ that the algorithm converges to. As shown in Figure 2, right panel, the algorithm converges to a well-conditioned factorization of $X^\star$, while the condition number is not forced to decrease when there is no regularizer.

7.3. Affine rank minimization using noiselet linear maps. In this task, we consider the problem of *affine rank minimization*, as described in section 1.1.1. We use permuted and

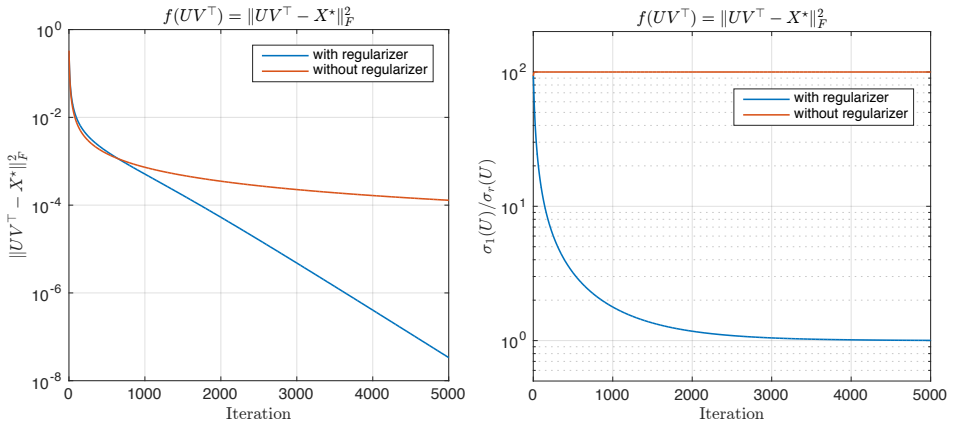

Figure 2. *$f(UV^\top) = \left\|UV^\top - X^\star\right\|_F^2$, where $X^\star = U^\star V^{\star\top} \in \mathbb{R}^{100\times 100}$, and $U^\star, V^\star \in \mathbb{R}^{100\times 10}$ are orthonormal. The initial point is an ill-conditioned pair ($\sigma_1(U_0)/\sigma_r(U_0) = \sigma_1(V_0)/\sigma_r(V_0) = 10^2$) near $X^\star$. Left panel: Convergence behaviors of* `BFGD` *with a regularizer $g = \frac{1}{4}\left\|U^\top U - V^\top V\right\|_F^2$ and without any regularizer. Right panel: The ratios $\sigma_1(U)/\sigma_r(U)$ over iterations.*

subsampled noiselets for the linear operator $\mathcal{A}$, due to their efficient implementation [99]; similar results can be obtained for $\mathcal{A}$ being a subsampled Fourier linear operator or, even, a random Gaussian linear operator. For the purposes of this experiment, the ground truth $X^\star$ is synthetically generated as the multiplication of two tall matrices, $U^\star \in \mathbb{R}^{m\times r}$ and $V^\star \in \mathbb{R}^{n\times r}$, such that $X^\star = U^\star V^{\star\top}$ and $\|X^\star\|_F = 1$. Both $U^\star$ and $V^\star$ contain random, independent and identically distributed (i.i.d.) Gaussian entries, with zero mean and unit variance.

List of algorithms. We compare the following state-of-the-art algorithms: (i) the singular value projection (`SVP`) algorithm [53] *constant* step size selection $\mu = 1/3$, as it is the one that showed the best performance in our experiments, (ii) the SparseApproxSDP extension to nonsquare cases for (4) in [51], based on [49], (iii) the matrix completion algorithm in [89], which we call `GuaranteedMC`[12], (iv) the Procrustes Flow algorithm in [92], and (v) the `BFGD` algorithm.[13]

Implementation details. In all experiments, we fix the number of observations in y to $p = C \cdot n \cdot r$, where $n \geq m$ in our cases, and for varying values of C. We use a MATLAB environment, where no `mex`-ified parts present, apart from those used in SVD calculations; see below.

We fix the maximum number of iterations to $T = 4000$, unless otherwise stated. We use the same stopping criteria for the majority of algorithms as $\frac{\|X_t - X_{t-1}\|_F}{\|X_t\|_F} \leq \text{tol}$, where X_t, X_{t-1} denote the current and the previous estimates in the X space and $\text{tol} := 5 \cdot 10^{-6}$. For SVD calculations, we use the `lansvd` implementation in PROPACK package [67]. For fairness, we modified all the algorithms so that they *exploit the true rank* r; however, we observed

[12] We note that the original algorithm in [89] is designed for the matrix *completion* problem, not the matrix *sensing* problem here.

[13] The algorithm in [106] assumes a step size that depends on RIP constants, which are NP-hard to compute; since no heuristic is proposed, we do not include this algorithm in the comparison list.

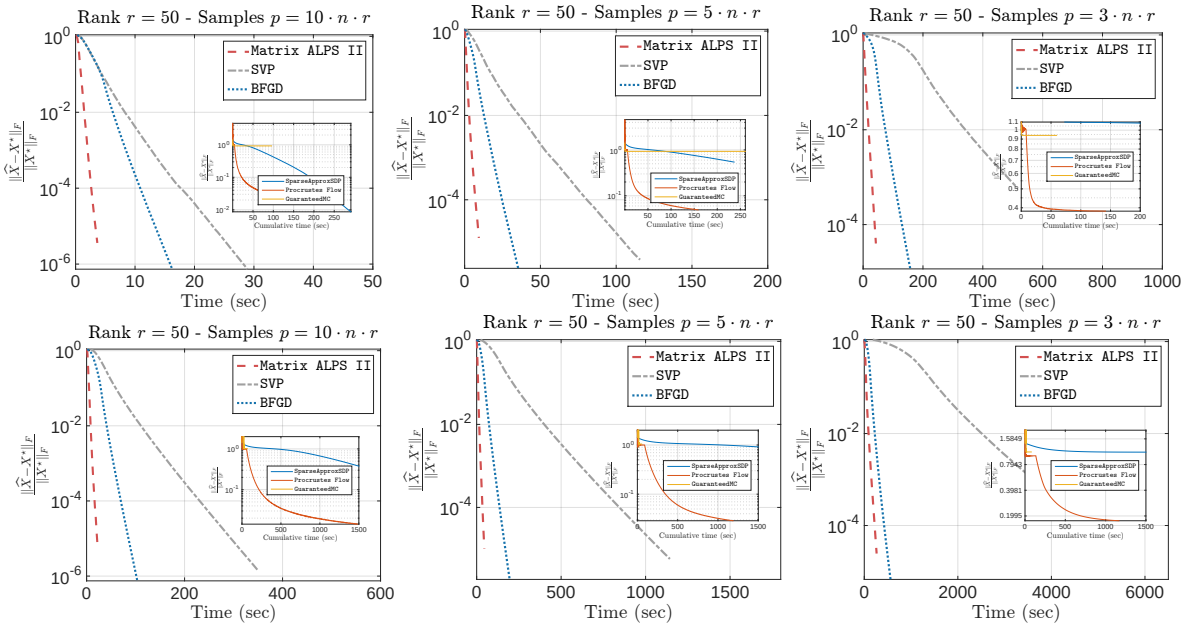

Figure 3. *Convergence performance of algorithms under comparison w.r.t.* $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$ *versus the total execution time. Top row corresponds to dimensions* $m = n = 1024$*; bottom row corresponds to dimensions* $m = 2048$, $n = 4096$. *Details on problem configuration are given on plots' title. For all cases, we used* $\mathcal{A}$ *as noiselets and* $r = 50$.

that small deviations from the true rank result in relatively small degradation in terms of the reconstruction performance.[14]

In the implementation of `BFGD`, we set g to be $\frac{1}{16} \cdot \|U^\top U - V^\top V\|_F^2$. Moreover, for our implementation of Procrustes Flow, we set the constant step size as $\mu := \frac{2}{187} \cdot \{\frac{1}{\|U_0\|_F^2}, \frac{1}{\|V_0\|_F^2\|}\}$, as suggested in [92]. We use the implementation of [89], with random initialization (unless otherwise stated) and regularization type `soft`, as suggested by their implementation. In [51], we require an upper bound on the nuclear norm of $X^\star$; in our experiments we assume we know $\|X^\star\|_*$, which requires a full SVD calculation. Moreover, we set the curvature constant for the SparseApproxSDP implementation to its true value $C_f = 1$.

For initialization, we consider the following settings: (i) random initialization, where $X_0 = U_0 V_0^\top$ for some randomly selected U_0 and V_0 such that $\|X_0\|_F = 1$, and (ii) specific initialization, as suggested in each of the papers above. Our specific initialization is based on the discussion in section 5, where $X_0 = \mathcal{P}_r(-\frac{1}{L}\nabla f(0))$. Algorithms SVP and SparseApprox-SDP and the solver in [89] work with random initialization. For the initialization phase of [92], we consider two cases: (i) the condition number κ is known, where according to Theorem 3.3 in [92], we require $T_{\text{init}} := \lceil 3\log(\sqrt{r}\cdot\kappa)+5 \rceil$ SVP iterations, and (ii) the condition number κ is unknown, where we use Lemma 3.4 in [92].

Results using random initialization. Figure 3 depicts the convergence performance of the above algorithms w.r.t. total execution time. `BFGD` shows the best performance, compared to

[14] In case the rank of $X^\star$ is unknown, one has to predict the dimension of the principal singular space. The authors in [53], based on ideas in [61, 59], propose to compute singular values incrementally until a significant gap between singular values is found.

Table 3

Summary of results for reconstruction and efficiency. Here, $m = n = 1024$, resulting in 1,048,576 variables to optimize, and $\mathcal{A}$ is a noiselet-based subsampled linear map. The number of samples p satisfies $p = C \cdot n \cdot r$ for various values of constant C. Time reported is in seconds.

	$r = 50$, $C = 10$		$r = 50$, $C = 5$		$r = 50$, $C = 3$	
Algorithm	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time
[53]	6.86e-07	29.05	3.75e-06	115.90	5.73e-04	517.56
[92]	8.65e-03	291.04	5.44e-01	236.44	1.08e+00	223.24
[51]	1.56e-02	223.15	4.92e-02	158.54	3.84e-01	141.59
[89]	9.25e-01	95.51	9.31e-01	260.84	9.39e-01	59.42
BFGD	7.08e-07	16.28	2.31e-06	35.39	1.15e-05	157.66

Table 4

Summary of results for reconstruction and efficiency. Here, $m = 2048$, $n = 4096$, resulting in 8,388,608 variables to optimize, and $\mathcal{A}$ is a noiselet-based subsampled linear map. The number of samples p satisfies $p = C \cdot n \cdot r$ for various values of constant C. Time reported is in seconds.

	$r = 50$, $C = 10$		$r = 50$, $C = 5$		$r = 50$, $C = 3$	
Algorithm	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time
[53]	1.41e-06	349.70	5.69e-06	1144.74	1.41e-04	4703.20
[92]	2.11e-01	1909.47	8.79e-01	1653.70	1.10e+00	1692.74
[51]	1.42e-02	1484.22	2.88e-02	1187.52	1.75e-01	1165.42
[89]	1.01e+00	69.22	1.04e+00	53.16	1.11e+00	78.23
BFGD	6.97e-07	103.83	1.79e-06	195.51	6.67e-06	561.82

the rest of the algorithms. It is notable that BFGD performs better than SVP, by avoiding SVD calculations and employing a better step size selection.[15] For this setting, GuaranteedMC converges to a local minimum, while SparseApproxSDP and Procrustes Flow show a close to sublinear convergence rate.

To further show how the performance of each algorithm scales as dimension increases, we provide aggregated results in Tables 3–5. Observe that BFGD is one order of magnitude faster than the rest of the nonconvex factorization algorithms. Observe that SVP requires one order of magnitude more time to complete one iteration, mostly due to the SVD step. In stark contrast, all factorization-based approaches spend less time per iteration, as was expected by the discussion in section 7.1.

Results using specific initialization.[16] In this case, we study the effect of initialization in the convergence performance of each algorithm. To do so, we focus only on the factorization-based algorithms: Procrustes Flow, GuaranteedMC, and BFGD. We consider two problem cases:

[15] If our step size is used in SVP, we get slightly better performance, but not in a universal manner.

[16] Bhojanapalli, Neyshabur, and Srebro [15] recently proved that random initialization is sufficient to lead to the optimum $X^\star$ for MS problems, while operating on the factors for $X^\star \succeq 0$. For the nonsquare case, see [84].

Table 5

Median time per iteration. Time reported is in seconds.

Algorithm	$m = n = 1024$, $C = 3$ Median time per iter.	$m = 2048$, $n = 4096$, $C = 3$ Median time per iter.
[53]	1.60e-01	1.04e+00
[92]	5.87e-02	4.52e-01
[51]	3.40e-02	3.00e-01
[89]	7.14e-02	4.05e-01
BFGD	5.33e-02	3.98e-01

Table 6

Summary of results of factorization algorithms using our proposed initialization.

Algorithm	$m = n = 1024$, $C = 10$, $r = 50$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$m = n = 1024$, $C = 10$, $r = 5$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time
[92]	2.27e+01	281.20	4.04e+01	192.09
[89]	9.25e-01	96.85	4.76e-01	2.47
BFGD	3.70e-06	52.52	8.12e-06	65.49

Table 7

Summary of results of factorization algorithms using each algorithm's proposed initialization.

Algorithm	$m = n = 1024$, $C = 10$, $r = 50$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$m = n = 1024$, $C = 10$, $r = 5$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time
[92]	3.29e-05	390.68	8.57e-04	2017.79
[89]	9.25e-01	114.93	1.01e+00	68.17
BFGD	3.69e-06	64.26	3.14e-06	74.23
Algorithm	$m = 2048, n = 4096$, $C = 10$, $r = 50$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time	$m = 2048, n = 4096$, $C = 10$, $r = 5$ $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F}$	Time
[92]	4.98e-02	265.27	4.22e-02	1497.68
[89]	4.76e-01	4.07	1.03e+00	35.05
BFGD	8.13e-06	83.34	5.84e-06	379.14

(i) all schemes use *our initialization procedure*, and (ii) each algorithm uses its own suggested initialization procedure. The results are depicted in Tables 6 and 7, respectively.

Using our initialization procedure for all algorithms, we observe that both Procrustes Flow and `GuaranteedMC` schemes can compute an approximation $\widehat{X}$ such that $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F} > 10^{-1}$. In contrast, our approach achieves a solution $\widehat{X}$ that is close to the stopping criterion, i.e., $\frac{\|\widehat{X}-X^\star\|_F}{\|X^\star\|_F} \approx 10^{-6}$.

Using different initialization schemes per algorithm, the results are depicted in Table 7. We recall that `GuaranteedMC` is designed for matrix completion tasks, where the linear operator is a selection mask of the entries. Observe that Procrustes Flow's performance improves significantly by using their proposed initialization: the idea is to perform SVP iterations to get to a good initial point; then switch to nonconvex factored gradient descent for low per-iteration complexity. However, this initialization is computationally expensive, as Table 7 indicates.

7.4. Image denoising as matrix completion problem. In this example, we consider the matrix completion setting for an image denoising task. In particular, we observe a limited number of pixels from the original image and perform a low-rank approximation based only on the set of measurements. We use real data images: while the true underlying image might not be low-rank, we apply our solvers to obtain low-rank approximations.

Figures 4–6 depict the reconstruction results for three image cases. In all cases, we compute the best 100-rank approximation of each image (see, e.g., the top middle image in Figure 4, where the full set of pixels is observed) and we observe only the 35% of the total number of pixels, randomly selected.

Our algorithm shows competitive performance compared to simple gradient descent schemes as `SVP` and Procrustes Flow, while being a fast and scalable solver. Table 8 contains timing results from 10 Monte Carlo random realizations for all image cases.

7.5. 1-bit matrix completion. For this task, we repeat the experiments in [34] and compare `BFGD` with their proposed schemes. We assume $X^\star \in \mathbb{R}^{m\times n}$ is an unknown low-rank matrix, satisfying $\|X^\star\|_\infty \leq \alpha$, $\alpha > 0$, from which we observe only a subset of indices $\Omega \subset [m] \times [n]$, according to the following rule:

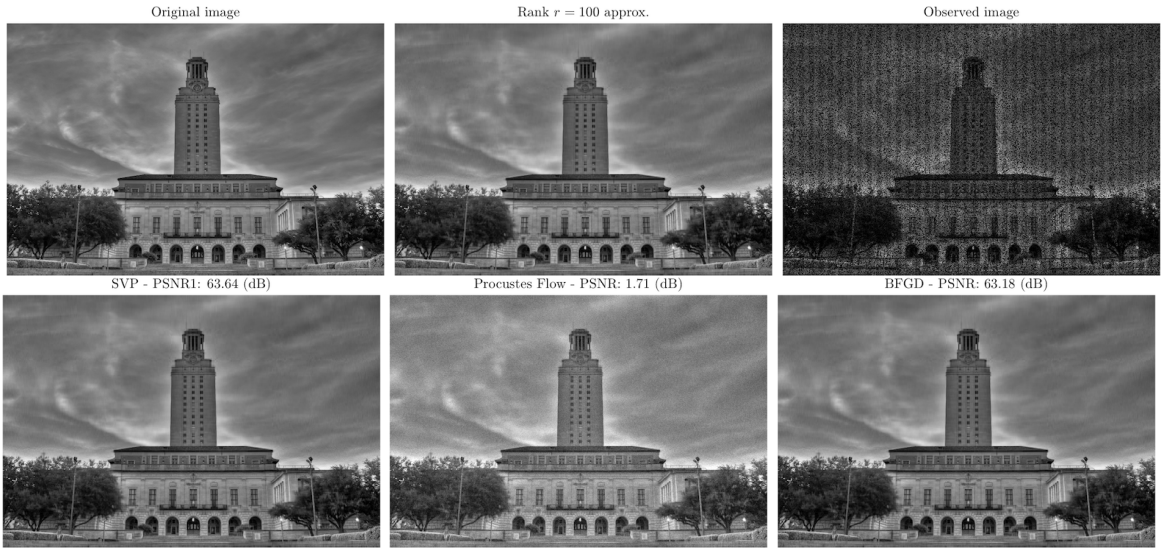

Figure 4. *Reconstruction performance in image denoising settings. The image size is* 2845×4266 *(*12,136,770 *pixels) and the approximation rank is preset to* $r = 100$. *We observe* 35% *of the pixels of the true image. We depict the median reconstruction error with respect to the true image in dB over* 10 *Monte Carlo realizations.*

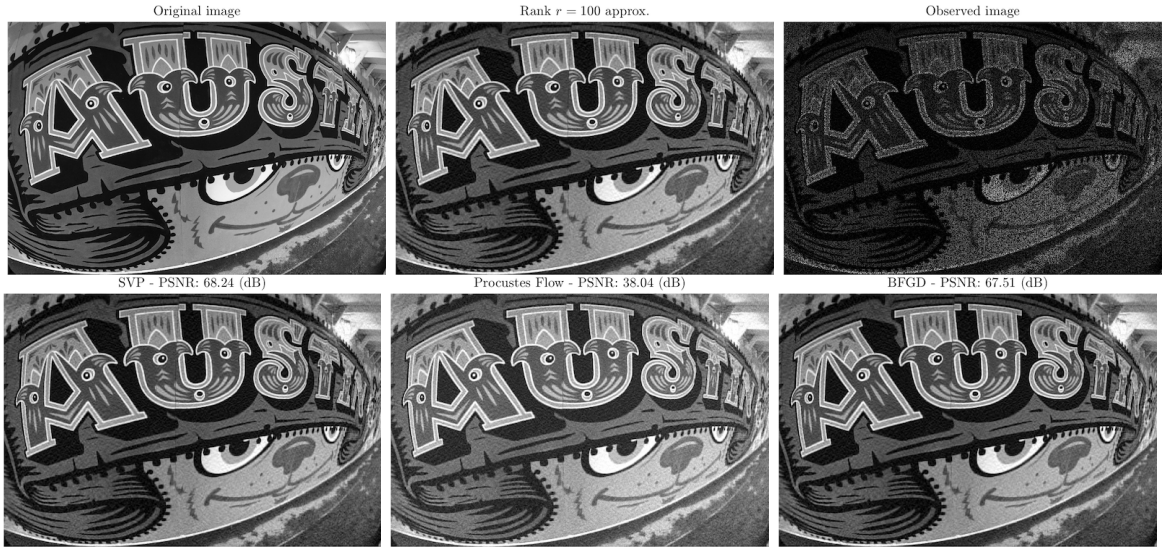

Figure 5. *Reconstruction performance in image denoising settings. The image size is* 3309×4963 *(*16,422,567 *pixels) and the approximation rank is preset to* $r = 100$*. We observe* 30% *of the pixels of the true image. We depict the median reconstruction error with respect to the true image in dB over* 10 *Monte Carlo realizations.*

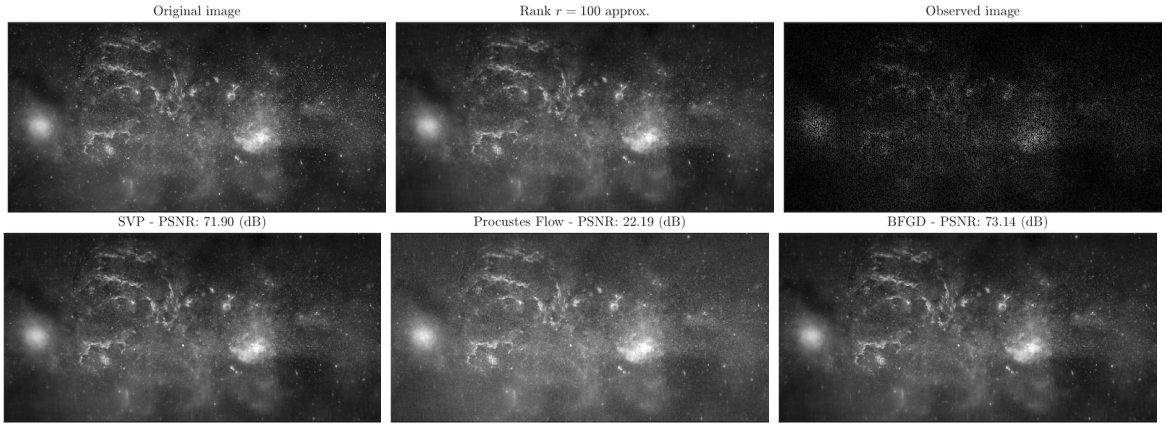

Figure 6. *Reconstruction performance in image denoising settings. The image size is* 4862×9725 *(*47,282,950 *pixels) and the approximation rank is preset to* $r = 100$*. We observe* 30% *of the pixels of the true image. We depict the median reconstruction error with respect to the true image in dB over* 10 *Monte Carlo realizations.*

$$Y_{i,j} = \begin{cases} +1 & \text{with probability } \sigma(X^\star_{i,j}) \\ -1 & \text{with probability } 1 - \sigma(X^\star_{i,j}) \end{cases} \qquad \text{for } (i,j) \in \Omega. \tag{21}$$

We assume Ω is chosen uniformly at random. Two natural choices for σ function are (i) the logistic regression model, where $\sigma(x) = \frac{e^x}{1+e^x}$, and (ii) the probit regression model, where

Table 8

Summary of execution time results for the problem of image denoising. Timings correspond to median values on 10 *Monte Carlo random instantiations.*

	Time (sec.)		
Algorithm	UT campus	Graffiti	Milky Way
[53]	5224.1	4154.9	7921.4
[92]	5383.4	6501.4	12806.3
BFGD	4062.4	3155.9	9119.6

$\sigma(x) = 1 - \Phi(-x/\sigma)$ for Φ being the cumulative Gaussian distribution function. Under this model, [34] propose two convex relaxation algorithmic solutions to recover $X^\star$: (i) the convex maximum log-likelihood estimator under nuclear norm and infinity norm constraints:

$$\begin{aligned} &\underset{X\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad f(X), \\ &\text{subject to} \quad \|X\|_* \le \alpha\sqrt{rmn}, \quad \|X\|_\infty \le \alpha, \end{aligned} \tag{22}$$

and (ii) the convex maximum log-likelihood estimator under only nuclear norm constraints. In both cases, $f(X)$ satisfies the expression in (6). Davenport et al. [34] propose a *spectral projected-gradient descent* method for both these criteria; in the case where only nuclear norm constraints are present, SVD routines compute the convex projection onto norm balls, while in the case where both nuclear and infinity norm constraints are present, [34] propose an alternating-direction method of multipliers solution.

Synthetic experiments. We synthetically construct $X^\star \in \mathbb{R}^{m\times n}$, where $m = n = 100$, such that $X^\star = U^\star V^{\star\top}$, where $U^\star \in \mathbb{R}^{m\times r}$, $V^\star \in \mathbb{R}^{n\times r}$ for $r = 1$. The entries of $U^\star$, $V^\star$ are drawn i.i.d. from Uni$[-\frac{1}{2}, \frac{1}{2}]$. According to [34], we scale $X^\star$ such that $\|X^\star\|_\infty = 1$. Then, we observe $Y \in \mathbb{R}^{m\times n}$ according to (21), where $|\Omega| = \frac{1}{4}\cdot mn$. We consider the probit regression model with additive Gaussian noise, with variance σ^2.

Figure 7 depicts the recovery performance of BFGD, as compared to variants of (22) in [34]. As noted in [34], the performance of all algorithms is poor when σ is too small or too large, while in between, for moderate noise levels, we observe better performance for all approaches.

By default, in all problem settings, we observe that the estimate of (22) *is not of low rank*: to compute the closest rank-r approximation to that, we further perform a *debias* step via truncated SVD. The effect of the debias step is better illustrated in Figure 7, focusing on the differences between the left and right plots: without such a step, BFGD has a better performance within the "sweet" range of noise levels, compared to the convex analogue in (22). Applying the debias step, both approaches have comparable performance, with that of (22) being slightly better.

Perhaps somewhat surprisingly, the performance of BFGD, in terms of *estimating the correct sign pattern* of the entries, is better than that of [34], even with the debias step. Figure 8, left panel, illustrates the performances for various noise levels.

Finally, we study the performance of the algorithms under consideration as a function of the number of measurements for fixed settings of dimensions $m = n = 200$ and noise level

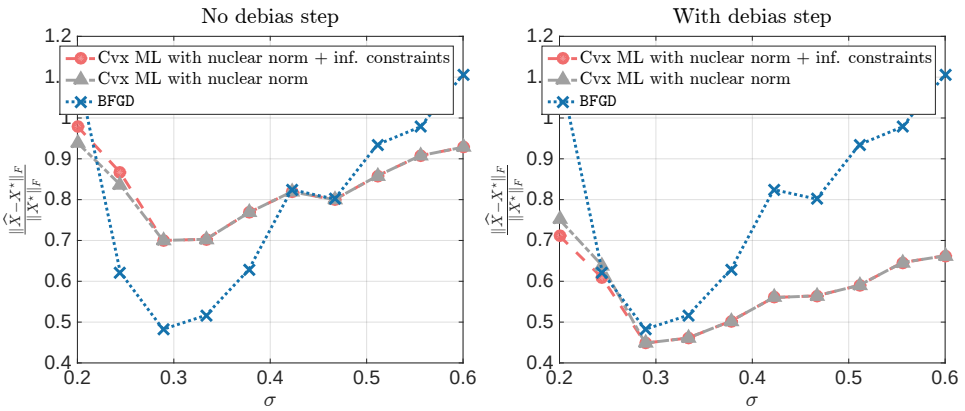

Figure 7. *Comparison of* 1*-bit matrix procedures. Left panel: Output of* (22) *is not projected onto rank-r set. Right panel: Output of* (22) *is projected onto rank-r set.*

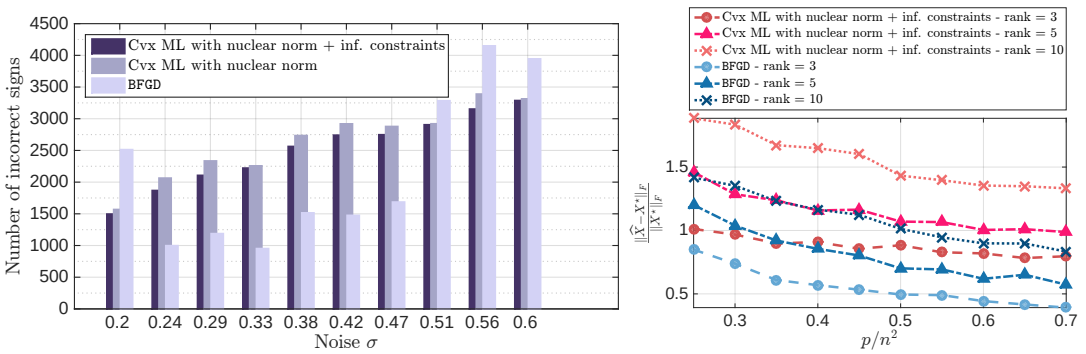

Figure 8. Left panel: *Comparison of* 1*-bit matrix procedures w.r.t. sign pattern estimation.* Right panel: *Recovery of* $X^\star$ *from* $p = C \cdot n^2$ *measurements.* $X^\star$ *is designed to be low rank:* $r = 3, 5,$ *and* 10. *x-axis represents* C *for various values.*

$\sigma = 0.244$. By the discussion above, such a noise level leads to good performance from all schemes. We considered matrices $X^\star$ with rank $r \in \{3,\ 5,\ 10\}$ and generate $p = C \cdot n^2$, over a wide range of $0 < C < 1$. Figure 8, right panel, shows the performance of `BFGD` and the approach for (22) in [34], in terms of the relative Frobenius norm of the error. All approaches do poorly when there are only $p < 0.35 \cdot n^2$ measurements, since this is near the noiseless information-theoretic limit. For higher numbers of measurements, the nonconvex approach in `BFGD` returns more reasonable solutions and outperforms convex approaches, taking advantage of the prior knowledge on low-rankness of the solution.

MovieLens data set. We compare 1-bit matrix completion solvers on the 100k MovieLens data set. To do so, we repeat the experiment in section 4.3 of [34]: we use the MovieLens 100k, which consists of 100K movie ratings, from 1000 users on 1700 movies. Each user entry denotes the movie rating, ranging from 1 to 5. To convert this data set into 1-bit measurements, we convert these ratings to binary observations by comparing each rating to the average rating for the entire data set (which is approximately 3.5), according to [34]. To evaluate the performance of the algorithms, we assume part of the observed ratings as unobserved (5K of

Table 9

Summary of results for the problem of 1-bit matrix completion on MovieLens data set. Individual and overall ratings correspond to percentages of signs correctly estimated (+1 corresponds to original rating above 3.5, −1 corresponds to original rating below 3.5). Timings correspond to median values on 10 Monte Carlo random instantiations.

Algorithm	Ratings (%)					Overall (%)	Time
	1	2	3	4	5		
SPG ($\alpha\sqrt{r} = 0.32$)	73.7	68.4	52.5	**74.9**	**91.0**	71.3	79.5
SPG ($\alpha\sqrt{r} = 4.64$)	77.2	71.0	58.5	72.5	86.9	71.8	213.4
SPG ($\alpha\sqrt{r} = 10.00$)	76.2	71.3	58.3	71.0	85.7	71.0	491.8
TFOCS	70.4	69.4	**59.2**	39.1	59.4	64.8	42.3
BFGD ($r = 3$)	**79.4**	74.5	56.9	72.5	88.2	**72.2**	**25.4**
BFGD ($r = 5$)	79.0	72.4	56.8	71.6	86.2	71.2	27.5
BFGD ($r = 10$)	77.6	**75.0**	57.5	70.5	84.1	70.9	30.3

them) and check if the estimate of $X^\star$, $\widehat{X}$, predicts the sign of these ratings. We perform machine learning estimation using logistic function $\sigma(x) = \frac{e^x}{1+e^x}$ in f.

We compare the following algorithms: (i) the SPG of (22) in [34] for 1-bit matrix completion, (ii) TFOCS [10], where we observe the *unquantized* data set (actual values), and (iii) BFGD for various values of rank parameter r. The results are shown Table 9 over 10 Monte Carlo realizations (i.e., we randomly selected 5K ratings as test sets 10 times and solved the problem). BFGD shows competitive performance, compared to convex approaches. Moreover, setting the parameter r is an "easier" and more intuitive task: our algorithm administers precise control on the rankness of the solution, which might lead to further interpretation of the results. Convex approaches lack this property: the mapping between the regularization parameters and the number of rank-1 components in the extracted solution is highly nonlinear. At the same time, BFGD shows faster convergence to a good solution, which makes it a preferable algorithmic solution for large-scale applications.

Appendix A. Connection of η and $\widehat{\eta}$.

Proof of (16)⇒(15). Let $R_t^\star$ be the $r \times r$ orthogonal matrix such that

$$\textsc{Dist}(U_t, V_t; X_r^\star) = \|W_t - W^\star R_t^\star\|_F .$$

By the triangle inequality, we have

$$\begin{aligned} \|W_t\|_2 = \|W_t - W^\star R_t^\star + W^\star R_t^\star\|_2 &\overset{\text{(i)}}{\leq} \|W^\star R_t^\star\|_2 + \|W_t - W^\star R_t^\star\|_2 \\ &\overset{\text{(ii)}}{\leq} \|W^\star\|_2 + \tfrac{\sqrt{2}\sigma_r(X_r^\star)^{1/2}}{10} \overset{\text{(iii)}}{\leq} \|W^\star\|_2 + \tfrac{\sigma_r(W^\star)}{10} \leq \tfrac{11}{10} \cdot \|W^\star\|_2, \end{aligned} \tag{23}$$

where (i) is due to the triangle inequality, (ii) is due to Assumption 1, and (iii) is due to the fact that $\sqrt{2} \cdot \sigma_r(X_r^\star)^{1/2} = \sigma_r(W^\star)$ and $\kappa \geq 1$. The above bound holds for every t.

On the other hand, we have:

$$\begin{aligned}\|W_0\|_2 &= \|W_0 - W^\star R_t^\star + W^\star R^\star\|_2 \overset{\text{(i)}}{\geq} \|W^\star R_t^\star\|_2 - \|W_0 - W^\star R_t^\star\|_2 \\ &\geq \|W^\star\|_2 - \tfrac{\sqrt{2}\sigma_r(X_r^\star)^{1/2}}{10} \geq \|W^\star\|_2 - \tfrac{\sigma_1(W^\star)}{10} \geq \tfrac{9}{10}\cdot\|W^\star\|_2. \end{aligned} \tag{24}$$

Combining (23) and (24), we obtain

$$\|W_t\|_2 \leq \tfrac{11}{10}\cdot\|W^\star\|_2 \leq \tfrac{11}{9}\|W_0\|_2 \Longrightarrow \tfrac{81}{121}\cdot\|W_t\|_2^2 \leq \|W_0\|_2^2$$

and finally $\frac{1}{8\cdot\max\{L,\, L_g\}\cdot\|W_t\|_2^2} = \frac{1}{8\cdot\frac{121}{81}\cdot\max\{L,\, L_g\}\cdot\frac{81}{121}\cdot\|W_t\|_2^2} \geq \frac{1}{12\cdot\max\{L,\, L_g\}\cdot\|W_0\|_2^2}$. ∎

Proof of (17)⇒(13). We have

$$\begin{aligned}\|\nabla f(U_tV_t^\top)\|_2 &\leq \|\nabla f(U_0V_0^\top)\|_2 + \|\nabla f(U_tV_t^\top) - \nabla f(U_0V_0^\top)\|_2 \\ &\overset{\text{(i)}}{\leq} \|\nabla f(U_0V_0^\top)\|_2 + L\|U_tV_t^\top - U_0V_0^\top\|_F \\ &\overset{\text{(ii)}}{\leq} \|\nabla f(U_0V_0^\top)\|_2 + L\|U_tV_t^\top - U^\star V^{\star\top}\|_F + L\|U_0V_0^\top - U^\star V^{\star\top}\|_F, \end{aligned} \tag{25}$$

where (i) is due to the fact that f is L-smooth and (ii) holds by adding and subtracting $U^\star V^{\star\top}$ and then applying the triangle inequality. To bound the last two terms on the right-hand side, we observe

$$\begin{aligned}\|U_tV_t^\top - U^\star V^{\star\top}\|_F &= \|U_tV_t^\top - U^\star RV_t^\top + U^\star RV_t^\top - U^\star RR^\top V^{\star\top}\|_F \\ &\overset{\text{(i)}}{\leq} \|U^\star R\|_2\cdot\|V_t - V^\star R\|_F + \|V_t\|_2\cdot\|U_t - U^\star R\|_F \\ &\leq (\|U^\star\|_2 + \|V_t\|_2)\cdot\textsc{Dist}(U_t, V_t; X_r^\star) \\ &\overset{\text{(ii)}}{\leq} \frac{21}{10}\cdot\|W^\star\|_2\cdot\frac{\sigma_r(W^\star)}{10} \leq \frac{7}{10}\cdot\|W_0\|_2^2, \end{aligned}$$

where (i) is due to the triangle and Cauchy–Schwarz inequalities, and (ii) is by Assumption 1 and (23). Similarly, one can show that $\|U_0V_0^\top - U^\star V^{\star\top}\|_F \leq \frac{7}{10}\cdot\|W_0\|_2^2$. Thus, (25) becomes

$$\|\nabla f(U_0V_0^\top)\|_2 \geq \|\nabla f(U_tV_t^\top)\|_2 - \frac{3L}{2}\|W_0\|_2^2. \tag{26}$$

Applying (23), (24), and the above bound, we obtain the desired result. ∎

Appendix B. Proof of Theorem 4.2. For clarity, we omit the subscript t and use $(U,\ V)$ to denote the current estimate and (U^+, V^+) the next estimate. Further, we denote $\nabla g \triangleq \nabla g(U^\top U - V^\top V)$, where the gradient is taken over both U and V. We denote the stacked matrices of (U, V) and their variants as follows:

$$W = \begin{bmatrix}U\\V\end{bmatrix},\ W^+ = \begin{bmatrix}U^+\\V^+\end{bmatrix},\ W^\star = \begin{bmatrix}U^\star\\V^\star\end{bmatrix}.$$

Observe that $W, W^+, W^\star \in \mathbb{R}^{(m+n)\times r}$. Then, the main recursion of BFGD in Algorithm 1 can be succinctly written as

$$W^+ = W - \widehat{\eta}\nabla_W(f + \tfrac{1}{2}g),$$

where

$$\nabla_W(f+\tfrac{1}{2}g) = \begin{bmatrix} \nabla_U f(UV^\top) + \frac{1}{2}\nabla_U g \\ \nabla_V f(UV^\top) + \frac{1}{2}\nabla_V g \end{bmatrix} = \begin{bmatrix} \nabla f(UV^\top)V + \frac{1}{2}U\nabla g \\ \nabla f(UV^\top)^\top U - \frac{1}{2}V\nabla g \end{bmatrix}.$$

In the above formulation, we use as regularization parameter λ of the g function, $\lambda = \frac{1}{2}$.

Our discussion below is based on Assumption 1, where

$$\textsc{Dist}(U, V; X_r^\star) \le \frac{\sqrt{2}\cdot\sigma_r(X_r^\star)^{1/2}}{10\sqrt{\kappa}} = \frac{\sigma_r(W^\star)}{10\sqrt{\kappa}} \tag{27}$$

holds for the current iterate. The last equality is due to the fact that $\sigma_r(W^\star) = \sqrt{2}\cdot\sigma_r(X_r^\star)^{1/2}$ for $(U^\star, V^\star)$ with "equal footing." For the initial point (U_0, V_0), (27) holds by the assumption of the theorem. Since the right-hand side is fixed, (27) holds for every iterate, as long as $\textsc{Dist}(U, V; X_r^\star)$ decreases.

To show this, let $R \in O_r$ be the minimizing orthogonal matrix such that

$$\textsc{Dist}(U, V; X_r^\star) = \|W - W^\star R\|_F;$$

here, O_r denotes the set of $r \times r$ orthogonal matrices such that $R^\top R = I$. Then, the decrease in distance can be lower bounded by

$$\begin{aligned} &\textsc{Dist}(U, V; X_r^\star)^2 - \textsc{Dist}(U^+, V^+; X_r^\star)^2 \\ &\qquad = \|W - W^\star R\|_F^2 - \min_{Q\in O_r} \left\|W^+ - W^\star Q\right\|_F^2 \\ &\qquad \ge \|W - W^\star R\|_F^2 - \left\|W^+ - W^\star R\right\|_F^2 \\ &\qquad = 2\widehat{\eta}\cdot\left\langle \nabla_W(f+\tfrac{1}{2}g), W - W^\star R\right\rangle - \widehat{\eta}^2\cdot\left\|\nabla_W(f+\tfrac{1}{2}g)\right\|_F^2, \end{aligned} \tag{28}$$

where the last equality is obtaining by substituting W^+, according to its definition above. To bound the first term on the right-hand side, we use the following lemma; the proof is provided in section B.1.

Lemma B.1 (descent lemma). *Let (27) hold for W. Let $\mu_{\min} = \min\{\mu, \mu_g\}$ and $L_{\max} = \max\{L, L_g\}$ for (μ, L) and (μ_g, L_g) the strong convexity and smoothness parameters pairs for f and g, respectively. Then, the following inequality holds:*

$$\begin{aligned} \left\langle \nabla_W(f+\tfrac{1}{2}g), W - W^\star R\right\rangle \ge{}& \tfrac{\mu_{\min}\cdot\sigma_r(W^\star)^2}{20}\|W - W^\star R\|_F^2 + \tfrac{1}{4L_{\max}}\left\|\nabla f(UV^\top)\right\|_F^2 \\ &+ \tfrac{1}{16L_{\max}}\|\nabla g\|_F^2 - \tfrac{L}{2}\|X^\star - X_r^\star\|_F^2. \end{aligned} \tag{29}$$

For the second term on the right-hand side of (28), we obtain the following upper bound:

$$
\begin{aligned}
\left\|\nabla_W(f+\tfrac{1}{2}g)\right\|_F^2 &= \left\|\begin{bmatrix}\nabla f(UV^\top)V+\frac{1}{2}U\nabla g\\ \nabla f(UV^\top)^\top U-\frac{1}{2}V\nabla g\end{bmatrix}\right\|_F^2\\
&= \left\|\nabla f(UV^\top)V+\tfrac{1}{2}U\nabla g\right\|_F^2+\left\|\nabla f(UV^\top)^\top U-\tfrac{1}{2}V\nabla g\right\|_F^2\\
&\overset{(a)}{\le} 2\left\|\nabla f(UV^\top)V\right\|_F^2+\tfrac{1}{2}\left\|U\nabla g\right\|_F^2+2\left\|\nabla f(UV^\top)^\top U\right\|_F^2+\tfrac{1}{2}\left\|V\nabla g\right\|_F^2\\
&= 2\left\|\nabla f(UV^\top)V\right\|_F^2+2\left\|\nabla f(UV^\top)^\top U\right\|_F^2+\tfrac{1}{2}\left\|W\nabla g\right\|_F^2\\
&\overset{(b)}{\le} 2\left\|\nabla f(UV^\top)\right\|_F^2\cdot\left(\|U\|_2^2+\|V\|_2^2\right)+\tfrac{1}{2}\|W\|_2^2\|\nabla g\|_F^2\\
&\overset{(c)}{\le} \left(4\left\|\nabla f(UV^\top)\right\|_F^2+\tfrac{1}{2}\|\nabla g\|_F^2\right)\cdot\|W\|_2^2,
\end{aligned}
\tag{30}
$$

where (a) follows from the fact $\|A+B\|_F^2\le 2\|A\|_F^2+2\|B\|_F^2$, (b) is due to the fact $\|AB\|_F\le\|A\|_F\cdot\|B\|_2$, and (c) follows from the observation that $\|U\|_2,\|V\|_2\le\|W\|_2$.

Plugging (29) and (30) into (28), we get

$$
\begin{aligned}
&\textsc{Dist}(U,V;X_r^\star)^2-\textsc{Dist}(U^+,V^+;X_r^\star)^2\\
&\qquad\ge 2\widehat{\eta}\cdot\left\langle\nabla_W(f+\tfrac{1}{2}g),W-W^\star R\right\rangle-\widehat{\eta}^2\cdot\left\|\nabla_W(f+\tfrac{1}{2}g)\right\|_F^2\\
&\qquad\ge \tfrac{\widehat{\eta}\cdot\mu_{\min}\cdot\sigma_r(W^\star)^2}{10}\textsc{Dist}(U,V;X_r^\star)^2-\widehat{\eta}L\left\|X^\star-X_r^\star\right\|_F^2\\
&\qquad= \tfrac{\widehat{\eta}\cdot\mu_{\min}\cdot\sigma_r(X_r^\star)}{5}\textsc{Dist}(U,V;X_r^\star)^2-\widehat{\eta}L\left\|X^\star-X_r^\star\right\|_F^2,
\end{aligned}
$$

where we use the fact that $\sigma_r(W^\star)=\sqrt{2}\cdot\sigma_r(X_r^\star)^{1/2}$.

The above lead to the following recursion:

$$
\textsc{Dist}(U^+,V^+;X_r^\star)^2\le\gamma_t\cdot\textsc{Dist}(U,V;X_r^\star)^2+\widehat{\eta}L\left\|X^\star-X_r^\star\right\|_F^2,
$$

where $\gamma_t=1-\frac{\widehat{\eta}\cdot\mu_{\min}\cdot\sigma_r(X_r^\star)}{5}$. By the definition of $\widehat{\eta}$ in (16), we further have

$$
\gamma_t=1-\tfrac{\mu_{\min}\cdot\sigma_r(X_r^\star)}{40\cdot L_{\max}\cdot\|W\|_2^2}\overset{(i)}{\ge}1-\tfrac{\mu_{\min}\cdot\sigma_r(X_r^\star)}{40\cdot L_{\max}\cdot\frac{81}{100}\cdot\|W^\star\|_2^2}\overset{(ii)}{\ge}1-\tfrac{\mu_{\min}}{65\cdot L_{\max}}\cdot\tfrac{\sigma_r(X_r^\star)}{\sigma_1(X_r^\star)},
$$

where (i) is by using (24) that connects $\|W\|_2$ with $\|W^\star\|_2$ as $\|W\|_2\ge\frac{9}{10}\|W^\star\|_2$ and (ii) is due to the fact $\|W^\star\|_2=\sqrt{2}\cdot\sigma_1(X_r^\star)^{1/2}$. ∎

B.1. Proof of Lemma B.1. Before we step into the proof, we require some more notation for simpler presentation of our ideas. We use another set of stacked matrices $Y=\left[\begin{smallmatrix}U\\-V\end{smallmatrix}\right]$, $Y^\star=\left[\begin{smallmatrix}U^\star\\-V^\star\end{smallmatrix}\right]$. The error of the current estimate from the closest optimal point is denoted by the following $\Delta_\times$ matrix structures:

$$
\Delta_U=U-U^\star R,\quad \Delta_V=V-V^\star R,\quad \Delta_W=W-W^\star R,\quad \Delta_Y=Y-Y^\star R.
$$

For our proof, we can write

$$\langle \nabla_W (f + \tfrac{1}{2}g), W - W^\star R \rangle = \underbrace{\langle \nabla f(UV^\top)V, U - U^\star R\rangle + \langle \nabla f(UV^\top)^\top U, V - V^\star R\rangle}_{(A)} + \tfrac{1}{2} \cdot \left(\underbrace{\langle U \nabla g, U - U^\star R\rangle - \langle V \nabla g, V - V^\star R\rangle}_{(B)} \right).$$

For (A), we have

$$\begin{aligned} (A) &= \langle \nabla f(UV^\top)V, U - U^\star R\rangle + \langle \nabla f(UV^\top)^\top U, V - V^\star R\rangle \\ &= \left\langle \nabla f(UV^\top), UV^\top - U^\star V^{\star\top} \right\rangle + \langle \nabla f(UV^\top), \Delta_U \Delta_V^\top \rangle \\ &\geq \frac{\mu}{2} \underbrace{\left\| UV^\top - U^\star V^{\star\top} \right\|_F^2}_{(A1)} + \frac{1}{2L} \underbrace{\left\| \nabla f(UV^\top) \right\|_F^2}_{(A2)} - \frac{L}{2} \underbrace{\| X^\star - X_r^\star \|_F^2}_{(A3)} \\ &\qquad - \underbrace{\left\| \nabla f(UV^\top) \right\|_2 \cdot \| \Delta_W \|_F^2}_{(A4)}, \end{aligned} \tag{31}$$

where, for the second term in (31), we use the fact that

$$\left\langle \nabla f(UV^\top), \Delta_U \Delta_V^\top \right\rangle \geq - \left| \left\langle \nabla f(UV^\top), \Delta_U \Delta_V^\top \right\rangle \right| = - \left| \left\langle \nabla f(UV^\top) \Delta_V, \Delta_U \right\rangle \right|,$$

the Cauchy–Schwarz inequality, and the fact that $\|\Delta_U\|_F, \|\Delta_V\|_F \leq \|\Delta_W\|_F$; the first term in (31) follows from

$$\begin{aligned} \left\langle \nabla f(UV^\top), UV^\top - U^\star V^{\star\top} \right\rangle &\overset{(i)}{\geq} f(UV^\top) - f(U^\star V^{\star\top}) + \frac{\mu}{2} \left\| UV^\top - U^\star V^{\star\top} \right\|_F^2 \\ &\overset{(ii)}{=} (f(UV^\top) - f(X^\star)) - (f(U^\star V^{\star\top}) - f(X^\star)) + \frac{\mu}{2} \left\| UV^\top - U^\star V^{\star\top} \right\|_F^2 \\ &\overset{(iii)}{\geq} \frac{1}{2L} \left\| \nabla f(UV^\top) \right\|_F^2 - \frac{L}{2} \left\| X^\star - U^\star V^{\star\top} \right\|_F^2 + \frac{\mu}{2} \left\| UV^\top - U^\star V^{\star\top} \right\|_F^2, \end{aligned}$$

where (i) is due to the μ-strong convexity of f, and (ii) is by adding and subtracting $f(X^\star)$; observe that $f(X^\star) = f(U^\star V^{\star\top})$ if and only if $\operatorname{rank}(X^\star) = r$, and (iii) is due to the L-smoothness of f and the fact that $\nabla f(X^\star) = 0$ (for the middle term) and due to the inequality [82, equation (2.1.7)] (for the first term):

$$f(X) + \langle \nabla f(X), Y - X \rangle + \tfrac{1}{2L} \cdot \|\nabla f(X) - \nabla f(Y)\|_F^2 \leq f(Y). \tag{32}$$

For (B), we have

$$\begin{aligned} (B) &= \langle Y \nabla g, W - W^\star R \rangle = \left\langle \nabla g, Y^\top W - Y^\top W^\star R \right\rangle \\ &= \tfrac{1}{2} \left\langle \nabla g, Y^\top W - R^\top Y^{\star\top} W^\star R \right\rangle + \tfrac{1}{2} \left\langle \nabla g, Y^\top W - 2 Y^\top W^\star R + R^\top Y^{\star\top} W^\star R \right\rangle \\ &\overset{(a)}{=} \tfrac{1}{2} \left\langle \nabla g, Y^\top W \right\rangle + \tfrac{1}{2} \left\langle \nabla g, Y^\top W - Y^\top W^\star R - R^\top Y^{\star\top} W + R^\top Y^{\star\top} W^\star R \right\rangle \end{aligned}$$

$$
\begin{aligned}
(33)\qquad &= \frac{1}{2}\left\langle \nabla g, U^\top U - V^\top V \right\rangle + \frac{1}{2}\left\langle \nabla g, \Delta_Y^\top \Delta_W \right\rangle \\
&\overset{(b)}{\geq} \frac{\mu_g}{4} \underbrace{\left\| U^\top U - V^\top V \right\|_F^2}_{(B1)} + \frac{1}{4L_g} \underbrace{\|\nabla g\|_F^2}_{(B2)} - \frac{1}{2} \underbrace{\|\nabla g\|_2 \cdot \|\Delta_W\|_F \cdot \|\Delta_Y\|_F}_{(B3)},
\end{aligned}
$$

where (a) follows from the "balance" assumption in $\mathcal{X}_r^\star$,

$$
Y^{\star\top} W^\star = U^{\star\top} U^\star - V^{\star\top} V^\star = 0,
$$

for the first term, and the fact that ∇g is symmetric, and therefore

$$
\left\langle \nabla g, Y^\top W^\star R \right\rangle = \left\langle \nabla g, R^\top W^{\star\top} Y \right\rangle = \left\langle \nabla g, R^\top Y^{\star\top} W \right\rangle
$$

for the second term; (b) follows from the fact

$$
\left\langle \nabla g, \Delta_Y^\top \Delta_W \right\rangle \geq - \left| \left\langle \nabla g, \Delta_Y^\top \Delta_W \right\rangle \right| = - \left| \langle \Delta_Y \nabla g, \Delta_W \rangle \right|,
$$

and the Cauchy–Schwarz inequality on the second term in (33), and

$$
\begin{aligned}
\left\langle \nabla g, U^\top U - V^\top V \right\rangle &\overset{(i)}{\geq} g(U^\top U - V^\top V) - g(0) + \tfrac{\mu_g}{2} \left\| U^\top U - V^\top V \right\|_F^2 \\
&\overset{(ii)}{\geq} \left\langle \nabla g(0), U^\top U - V^\top V \right\rangle + \tfrac{1}{2L_g} \|\nabla g - \nabla g(0)\|_F^2 + \tfrac{\mu_g}{2} \left\| U^\top U - V^\top V \right\|_F^2 \\
&\overset{(iii)}{=} \tfrac{1}{2L_g} \|\nabla g\|_F^2 + \tfrac{\mu_g}{2} \left\| U^\top U - V^\top V \right\|_F^2,
\end{aligned}
$$

where (i) follows from the strong convexity, (ii) is due to (32), and (iii) is by construction of g where $\nabla g(0) = 0$. Furthermore, (B1) can be bounded below as follows:

$$
\begin{aligned}
(B1) &= \left\| U^\top U - V^\top V \right\|_F^2 = \left\| U^\top U \right\|_F^2 + \left\| V^\top V \right\|_F^2 - 2\left\langle U^\top U, V^\top V \right\rangle \\
&= \left\| U U^\top \right\|_F^2 + \left\| V V^\top \right\|_F^2 - 2\left\langle U V^\top, U V^\top \right\rangle \\
&= \left\langle W W^\top, Y Y^\top \right\rangle \\
&= \left\langle W W^\top - W^\star W^{\star\top}, Y Y^\top - Y^\star Y^{\star\top} \right\rangle + \left\langle W^\star W^{\star\top}, Y Y^\top \right\rangle \\
&\qquad + \left\langle W W^\top - W^\star W^{\star\top}, Y^\star Y^{\star\top} \right\rangle \\
&\overset{(i)}{=} \left\langle W W^\top - W^\star W^{\star\top}, Y Y^\top - Y^\star Y^{\star\top} \right\rangle + \left\langle W^\star W^{\star\top}, Y Y^\top \right\rangle + \left\langle W W^\top, Y^\star Y^{\star\top} \right\rangle \\
&\geq \left\langle W W^\top - W^\star W^{\star\top}, Y Y^\top - Y^\star Y^{\star\top} \right\rangle \\
&= \left\| U U^\top - U^\star U^{\star\top} \right\|_F^2 + \left\| V V^\top - V^\star V^{\star\top} \right\|_F^2 - 2\left\| U V^\top - U^\star V^{\star\top} \right\|_F^2,
\end{aligned}
$$

where (i) is due to the fact that

$$\left\langle W^\star W^{\star\top}, Y^\star Y^{\star\top} \right\rangle = \left\| Y^{\star\top} W^\star \right\|_F^2 = \left\| U^{\star\top} U^\star - V^{\star\top} V^\star \right\|_F^2 = 0$$

and the first inequality holds by the fact that the inner product of two PSD matrices is nonnegative.

At this point, we have all the required components to compute the desired lower bound. Combining (A1) and (B1), we get

$$\begin{aligned} 4(A1) + (B1) &= \left\| UU^\top - U^\star U^{\star\top} \right\|_F^2 + \left\| VV^\top - V^\star V^{\star\top} \right\|_F^2 + 2 \left\| UV^\top - U^\star V^{\star\top} \right\|_F^2 \\ &= \left\| WW^\top - W^\star W^{\star\top} \right\|_F^2 \geq \frac{4\sigma_r(W^\star)^2}{5} \left\| \Delta_W \right\|_F^2, \end{aligned}$$

where, in order to obtain the last inequality, we borrow the following lemma by [92].

Lemma B.2. *For any $W, W^\star \in R^{(m+n)\times r}$, with $\Delta_W = W - W^\star R$ for some orthogonal matrix $R \in \mathbb{R}^{r\times r}$, we have*

$$\left\| WW^\top - W^\star W^{\star\top} \right\|_F^2 \geq 2 \cdot \left(\sqrt{2} - 1 \right) \cdot \sigma_r(W^\star)^2 \cdot \left\| \Delta_W \right\|_F^2.$$

For convenience, we further lower bound the right-hand side of this lemma by $2 \cdot (\sqrt{2} - 1) \cdot \sigma_r(W^\star)^2 \cdot \|\Delta_W\|_F^2 \geq \frac{4\sigma_r(W^\star)^2}{5} \|\Delta_W\|_F^2$.

Given the definitions of $\mu_{\min}$ and $L_{\max}$, we have

$$\begin{aligned} (A) + \tfrac{1}{2}(B) &\geq \tfrac{\mu}{2}(A1) + \tfrac{1}{2L}(A2) - \tfrac{L}{2}(A3) - (A4) + \tfrac{\mu_g}{8}(B1) + \tfrac{1}{8L_g}(B2) - \tfrac{1}{4}(B3) \\ &\overset{(i)}{\geq} \tfrac{\mu_{\min}}{8}(4(A1)+(B1)) + \tfrac{1}{2L_{\max}}(A2) + \tfrac{1}{8L_{\max}}(B2) - (A4) - \tfrac{1}{4}(B3) - \tfrac{L}{2}(A3) \\ &\geq \tfrac{\mu_{\min}\cdot\sigma_r(W^\star)^2}{10} \left\| \Delta_W \right\|_F^2 + \tfrac{1}{2L_{\max}} \left\| \nabla f(UV^\top) \right\|_F^2 + \tfrac{1}{8L_{\max}} \left\| \nabla g \right\|_F^2 \\ &\quad - \left\| \nabla f(UV^\top) \right\|_2 \left\| \Delta_W \right\|_F^2 - \tfrac{1}{4} \left\| \nabla g \right\|_F \left\| \Delta_W \right\|_F \left\| \Delta_Y \right\|_F \\ &\quad - \tfrac{L}{2} \left\| X^\star - X_r^\star \right\|_F^2, \end{aligned} \tag{34}$$

where in (i) we used the definitions of $\mu_{\min}$ and $L_{\max}$. Note that we have not used the condition (27). It follows from (27) that

$$\begin{aligned} \left\| \nabla f(UV^\top) \right\|_2 \cdot \left\| \Delta_W \right\|_F^2 &\leq \tfrac{\sigma_r(W^\star)}{10\sqrt{\kappa}} \left\| \nabla f(UV^\top) \right\|_2 \cdot \left\| \Delta_W \right\|_F \\ &\leq \tfrac{\mu_{\min}\cdot\sigma_r(W^\star)^2}{25} \left\| \Delta_W \right\|_F^2 + \tfrac{1}{4L_{\max}} \left\| \nabla f(UV^\top) \right\|_2^2 \end{aligned} \tag{35}$$

and

$$\begin{aligned} \tfrac{1}{4} \left\| \nabla g \right\|_2 \cdot \left\| \Delta_W \right\|_F \cdot \left\| \Delta_Y \right\|_F &= \tfrac{1}{4} \left\| \nabla g \right\|_2 \cdot \left\| \Delta_W \right\|_F^2 \\ &\leq \tfrac{\sigma_r(W^\star)}{40\sqrt{\kappa}} \left\| \nabla g \right\|_2 \cdot \left\| \Delta_W \right\|_F \\ &\leq \tfrac{\mu_{\min}\cdot\sigma_r(W^\star)^2}{100} \left\| \Delta_W \right\|_F^2 + \tfrac{1}{16L_{\max}} \left\| \nabla g \right\|_2^2, \end{aligned} \tag{36}$$

where we use the arithmetic mean-geometric mean inequality. Plugging (35) and (36) into (34), it is easy to obtain

$$(A) + \tfrac{1}{2}(B) \geq \tfrac{\mu_{\min}\cdot\sigma_r(W^\star)^2}{20}\left\|\Delta_W\right\|_F^2 + \tfrac{1}{4L_{\max}}\left\|\nabla f(UV^\top)\right\|_F^2 + \tfrac{1}{16L_{\max}}\left\|\nabla g\right\|_F^2 - \tfrac{L}{2}\left\|X^\star - X_r^\star\right\|_F^2. \qquad \blacksquare$$

Appendix C. Proof of Theorem 4.4. The proof follows the same framework of the sublinear convergence proof in [14]. We use the following general lemma to prove the sublinear converegence.

Lemma C.1. *Suppose that a sequence of iterates $\{W_t\}_{t=0}^T$ satisfies the following conditions:*

$$f(W_tW_t^\top) - f(W_{t+1}W_{t+1}^\top) \geq \alpha\cdot\left\|\nabla_W f(W_tW_t^\top)\right\|_F^2, \tag{37}$$

$$f(W_tW_t^\top) - f(W^\star W^{\star\top}) \leq \beta\cdot\left\|\nabla_W f(W_tW_t^\top)\right\|_F \tag{38}$$

for all $t = 0,\ldots,T-1$ and some values $\alpha,\beta > 0$ independent of the iterates. Then it is guaranteed that

$$f(W_TW_T^\top) - f(W^\star W^{\star\top}) \leq \frac{\beta^2}{\alpha\cdot T}.$$

Proof. Define $\delta_t = f(W_tW_t^\top) - f(W^\star W^{\star\top})$. If we get $\delta_{T_0} \leq 0$ at some $T_0 < T$, the desired inequality holds because the first hypothesis guarantees $\{\delta_t\}_{t=0}^T$ to be nonincreasing. Hence, we can only consider the time t, where $\delta_t, \delta_{t+1} \geq 0$. We have

$$\delta_{t+1} \overset{\text{(a)}}{\leq} \delta_t - \alpha\cdot\left\|\nabla_W f(W_tW_t^\top)\right\|_F^2 \overset{\text{(b)}}{\leq} \delta_t - \frac{\alpha}{\beta^2}\cdot\delta_t^2 \overset{\text{(c)}}{\leq} \delta_t - \frac{\alpha}{\beta^2}\cdot\delta_t\cdot\delta_{t+1},$$

where (a) follows from the first hypothesis, (b) follows from the second hypothesis, and (c) follows from that $\delta_{t+1} \leq \delta_t$ by the first hypothesis. Dividing by $\delta_t\cdot\delta_{t+1}$, we obtain

$$\frac{1}{\delta_{t+1}} - \frac{1}{\delta_t} \geq \frac{\alpha}{\beta^2}.$$

Then we obtain the desired result by telescoping the above inequality. ∎

Now it suffices to show `BFGD` provides a sequence $\{W_t\}_{t=0}^T$ satisfying the hypotheses of Lemma C.1.

Obtaining (37). Although f is nonconvex over the factor space, it is reasonable to obtain a new estimate (with a carefully chosen steplength) which is no worse than the current one, because the algorithm takes a gradient step.

Lemma C.2. *Let f be an L-smooth convex function. Let $X = WW^\top$ and $X^+ = W^+W^{+\top}$ be two consecutive estimates of `BFGD`. Then*

$$f(WW^\top) - f(W^+W^{+\top}) \geq \frac{3\eta}{5}\cdot\left\|\nabla_W f(WW^\top)\right\|_F^2. \tag{39}$$

Since we can fix the steplength η based on the initial solution so that it is independent of the following iterates, we have obtained the first hypothesis of Lemma C.1.

Obtaining (38). Consider the following assumption:

$$(\mathrm{A}): \quad \textsc{Dist}(U,V;X_r^\star) = \min_{R\in O(r)} \|W - W^\star R\|_F \le \frac{\sigma_r(W^\star)}{10}.$$

Trivially (A) holds for U_0 and V_0. Now we provide key lemmas, and then the convergence proof will be presented.

Lemma C.3 (suboptimality bound). *Assume that* (A) *holds for* W. *Then*

$$f(WW^\top) - f(W^\star W^{\star\top}) \le \frac{7}{3}\cdot \left\|\nabla_W f(WW^\top)\right\|_F \cdot \textsc{Dist}(U,V;X_r^\star).$$

Lemma C.4 (descent in distance). *Assume that* (A) *holds for* W. *If*

$$f(W^+W^{+\top}) \ge f(W^\star W^{\star\top}), \quad \textit{then} \quad \textsc{Dist}(U^+,V^+;X_r^\star) \le \textsc{Dist}(U,V;X_r^\star).$$

Combining the above two lemmas, we obtain

$$f(WW^\top) - f(W^\star W^{\star\top}) \le \frac{7\cdot \textsc{Dist}(U_0,V_0;X_r^\star)}{3}\cdot \left\|\nabla_W f(WW^\top)\right\|_F. \tag{40}$$

Plugging (39) and (40) into Lemma C.1, we obtain the desired result. ∎

C.1. Proof of Lemma C.2. The L-smoothness gives

$$\begin{aligned} &f(WW^\top) - f(W^+W^{+\top}) \\ &\ge \left\langle \nabla f(WW^\top), WW^\top - W^+W^{+\top}\right\rangle - \frac{L}{2}\left\|WW^\top - W^+W^{+\top}\right\|_F^2 \\ &= \left\langle \nabla f(WW^\top), (W-W^+)W^\top + W(W-W^+)^\top\right\rangle \\ &\qquad - \left\langle \nabla f(WW^\top), (W-W^+)(W-W^+)^\top\right\rangle \\ &\qquad - \frac{L}{2}\left\|WW^\top - W^+W^{+\top}\right\|_F^2. \end{aligned} \tag{41}$$

For the first term, we have

$$\begin{aligned} \left\langle \nabla f(WW^\top), (W-W^+)W^\top + W(W-W^+)^\top\right\rangle &= 2\cdot\left\langle \nabla f(WW^\top)W, W-W^+\right\rangle \\ &= \eta\cdot\left\|\nabla_W f(WW^\top)\right\|_F^2. \end{aligned} \tag{42}$$

Using the Cauchy–Schwarz inequality, the second term can be bounded as follows:

$$\begin{aligned} &\left\langle \nabla f(WW^\top), (W-W^+)(W-W^+)^\top\right\rangle \\ &= \eta^2\left\langle \nabla f(WW^\top), \nabla_W f(WW^\top)\cdot\nabla_W f(WW^\top)^\top\right\rangle \\ &= \eta^2\left\langle \nabla f(WW^\top)\cdot\nabla_W f(WW^\top), \nabla_W f(WW^\top)\right\rangle \\ &\le \eta^2\left\|\nabla f(WW^\top)\cdot\nabla_W f(WW^\top)\right\|_F\left\|\nabla_W f(WW^\top)\right\|_F \\ &\le \eta^2\left\|\nabla f(WW^\top)\right\|_2\left\|\nabla_W f(WW^\top)\right\|_F^2. \end{aligned} \tag{43}$$

To bound the third term of (41), we have

$$\begin{aligned}\|WW^\top - W^+W^{+\top}\|_F &\le \|WW^\top - WW^{+\top}\|_F + \|WW^{+\top} - W^+W^{+\top}\|_F \\ &\le (\|W\|_2 + \|W^+\|_2) \cdot \|W - W^+\|_F \\ &\le \eta \cdot \left(2\|W\|_2 + \eta \cdot \left\|\nabla f(WW^\top)\right\|_2 \cdot \|W\|_2\right) \cdot \|\nabla_W f(WW^\top)\|_F \\ &\le \frac{7\eta}{3}\|W\|_2 \cdot \|\nabla_W f(WW^\top)\|_F. \end{aligned} \tag{44}$$

Plugging (42), (43), and (44) into (41), we obtain

$$\begin{aligned} f(WW^\top) - f(W^+W^{+\top}) &\ge \eta \cdot \left\|\nabla_W f(UV^\top)\right\|_F^2 \cdot \left(1 - \eta\frac{17L\|W\|_2^2 + 3\|\nabla f(WW^\top)\|_2}{3}\right) \\ &\ge \frac{3\eta}{5} \cdot \left\|\nabla_W f(UV^\top)\right\|_F^2, \end{aligned}$$

where the last inequality follows from the condition of the steplength η. This completes the proof. ∎

C.2. Proof of Lemma C.3. We use the following lemma.

Lemma C.5 (error bound). *Assume that* (A) *holds for* W. *Then*

$$\left\langle \nabla f(WW^\top), \Delta_W \Delta_W^\top \right\rangle \le \frac{1}{3} \cdot \left\|\nabla_W f(UV^\top)\right\|_F \cdot \textsc{Dist}(U, V; X_r^\star).$$

Now the lemma is proved as follows:

$$\begin{aligned} & f(WW^\top) - f(W^\star W^{\star\top}) \\ &\overset{(a)}{\le} \langle \nabla f(WW^\top), WW^\top - W^\star W^{\star\top}\rangle \\ &= \langle \nabla f(WW^\top), \Delta_W W^\top\rangle + \langle \nabla f(WW^\top), W\Delta_W^\top\rangle - \langle \nabla f(WW^\top), \Delta_W\Delta_W^\top\rangle \\ &= 2\langle \nabla f(WW^\top)W, \Delta_W\rangle - \langle \nabla f(WW^\top), \Delta_W\Delta_W^\top\rangle \\ &\overset{(b)}{\le} 2 \cdot \left\|\nabla_W f(WW^\top)\right\|_F \cdot \|\Delta_W\|_F + |\langle \nabla f(WW^\top), \Delta_W\Delta_W^\top\rangle| \\ &\overset{(c)}{\le} \frac{7}{3} \cdot \left\|\nabla_W f(WW^\top)\right\|_F \cdot \|\Delta_W\|_F, \end{aligned}$$

where (a) follows from the convexity of f, (b) follows from the Cauchy–Schwarz inequality, and (c) follows from Lemma C.5. ∎

C.3. Proof of Lemma C.5. Define Q_W, $Q_{W^\star}$, and Q_{Δ_W} as the projection matrices of the column spaces of W, $W^\star$, and $\Delta_W = W - W^\star R$, respectively. We have

$$\begin{aligned} \left\langle \nabla f(WW^\top), \Delta_W\Delta_W^\top \right\rangle &= \left\langle \nabla f(WW^\top)Q_{\Delta_W}, \Delta_W\Delta_W^\top \right\rangle \\ &\overset{(a)}{\le} \left\|\nabla f(WW^\top)Q_{\Delta_W}\right\|_2 \cdot \|\Delta_W\|_F^2 \\ &\overset{(b)}{\le} \left(\left\|\nabla f(WW^\top)Q_W\right\|_2 + \left\|\nabla f(WW^\top)Q_{W^\star}\right\|_2\right) \cdot \|\Delta_W\|_F^2, \end{aligned} \tag{45}$$

where (a) follows from the Cauchy–Schwarz inequality and the fact $\|AB\|_F \le \|A\|_2 \cdot \|B\|_F$, and (b) follows from that $W - W^\star$ lies on the column space spanned by W and $W^\star$. To bound the terms in (45), we obtain

$$\begin{aligned}
\left\|\nabla f(WW^\top)Q_W\right\|_2 &= \left\|\nabla f(WW^\top)WW^\dagger\right\|_2 \le \frac{1}{\sigma_r(W)}\left\|\nabla f(WW^\top)W\right\|_2 \\
&\le \frac{10}{9\sigma_r(W^\star)}\left\|\nabla f(WW^\top)W\right\|_2, \\
\left\|\nabla f(WW^\top)Q_{W^\star}\right\|_2 &= \left\|\nabla f(WW^\top)W^\star W^{\star\dagger}\right\|_2 \le \frac{1}{\sigma_r(W^\star)}\left\|\nabla f(WW^\top)W^\star\right\|_2, \\
\|\nabla f(WW^\top)W^\star\|_2 &\le \|\nabla f(WW^\top)W\|_2 + \|\nabla f(WW^\top)\Delta_W\|_2 \\
&\le \|\nabla f(WW^\top)W\|_2 \\
&\qquad + \left(\|\nabla f(WW^\top)Q_W\|_2 + \|\nabla f(WW^\top)Q_{W^\star}\|_2\right)\cdot\|\Delta_W\|_2 \\
&\le \frac{10}{9}\|\nabla f(WW^\top)W\|_2 + \frac{1}{10}\cdot\|\nabla f(WW^\top)W^\star\|_2, \\
\|\nabla f(WW^\top)Q_{W^\star}\|_2 &\le \frac{1}{\sigma_r(W^\star)}\|\nabla f(WW^\top)W^\star\|_2 \le \frac{5}{4\sigma_r(W^\star)}\|\nabla f(WW^\top)W\|_2,
\end{aligned}$$

where $W^\dagger$ and $W^{\star\dagger}$ are the Moore–Penrose pseudoinverses of W and $W^\star$. Plugging the above into (45), we get

$$\begin{aligned}
\left\langle \nabla f(WW^\top), \Delta_W\Delta_W^\top\right\rangle &\le \frac{95}{36\sigma_r(W^\star)}\cdot\left\|\nabla f(WW^\top)W\right\|_2\cdot\|\Delta_W\|_F^2 \\
&\overset{(a)}{\le} \frac{1}{3}\cdot\left\|\nabla f(WW^\top)W\right\|_2\cdot\|\Delta_W\|_F,
\end{aligned}$$

where (a) follows from (A). ∎

C.4. Proof of Lemma C.4. For this proof, we borrow a lemma from [14]. Although the assumption for the lemma is stronger than assumption (A), a slight modification of the proof leads to the following lemma from assumption (A).

Lemma C.6 (Lemma C.2 of [14]). *Let assumption* (A) *hold and* $f(W^+W^{+\top}) \ge f(W^\star W^{\star\top})$. *Then the following lower bound holds:*

$$\left\langle \nabla f(WW^\top), \Delta_W\Delta_W^\top\right\rangle \ge -\frac{\sqrt{2}}{\sqrt{2}-\frac{1}{10}}\cdot\frac{1}{10}\cdot\left|\left\langle \nabla f(WW^\top), WW^\top - W^\star W^{\star\top}\right\rangle\right|.$$

We have

$$\begin{aligned}
&\textsc{Dist}(U,V;X_r^\star)^2 - \textsc{Dist}(U^+,V^+;X_r^\star)^2 \\
&\qquad \ge \|W - W^\star R\|_F^2 - \left\|W^+ - W^\star R\right\|_F^2 \\
&\qquad = 2\eta\left\langle \nabla_W f(WW^\top), \Delta_W\right\rangle - \eta^2\left\|\nabla_W f(WW^\top)\right\|_F^2 \\
&\qquad = 4\eta\left\langle \nabla f(WW^\top)W, \Delta_W\right\rangle - \eta^2\left\|\nabla_W f(WW^\top)\right\|_F^2
\end{aligned}$$

$$
\begin{aligned}
&= 2\eta \left\langle \nabla f(WW^\top), WW^\top - W^\star W^{\star\top} \right\rangle + 2\eta \left\langle \nabla f(WW^\top), \Delta_W \Delta_W^\top \right\rangle \\
&\qquad - \eta^2 \left\| \nabla_W f(WW^\top) \right\|_F^2 \\
&\overset{(a)}{\geq} \frac{17\eta}{10} \left\langle \nabla f(WW^\top), WW^\top - W^\star W^{\star\top} \right\rangle - \eta^2 \left\| \nabla_W f(WW^\top) \right\|_F^2 \\
&\overset{(b)}{\geq} \frac{51\eta^2}{50} \left\| \nabla_W f(WW^\top) \right\|_F^2 - \eta^2 \left\| \nabla_W f(WW^\top) \right\|_F^2 \geq 0, \qquad (46)
\end{aligned}
$$

where (a) follows from Lemma C.6, (b) follows from the convexity of f, the hypothesis of the lemma, and Lemma C.2 as follows:

$$
\begin{aligned}
\left\langle \nabla f(WW^\top), WW^\top - W^\star W^{\star\top} \right\rangle &\geq f(WW^\top) - f(W^\star W^{\star\top}) \\
&\geq f(WW^\top) - f(W^+ W^{+\top}) \\
&\geq \frac{3\eta}{5} \cdot \left\| \nabla_W f(WW^\top) \right\|_F^2 .
\end{aligned}
$$

This completes the proof. ∎

Appendix D. Initialization proofs. The triangle inequality gives that

$$
\left\| U_0 V_0^\top - X_r^\star \right\|_F \leq \left\| U_0 V_0^\top - X_0 \right\|_F + \|X_0 - X^\star\|_F + \|X^\star - X_r^\star\|_F . \qquad (47)
$$

Let us first obtain an upper bound on the first term. We have

$$
\begin{aligned}
\left\| X_0 - U_0 V_0^\top \right\|_F &= \left\| \begin{bmatrix} \sigma_{r+1}(X_0) \\ \vdots \\ \sigma_{\min\{m,n\}}(X_0) \end{bmatrix} \right\| \\
&\overset{(a)}{\leq} \left\| \begin{bmatrix} \sigma_{r+1}(X^\star) \\ \vdots \\ \sigma_{\min\{m,n\}}(X^\star) \end{bmatrix} \right\| + \left\| \begin{bmatrix} \sigma_{r+1}(X_0) - \sigma_{r+1}(X^\star) \\ \vdots \\ \sigma_{\min\{m,n\}}(X_0) - \sigma_{\min\{m,n\}}(X^\star) \end{bmatrix} \right\| \\
&= \|X^\star - X_r^\star\|_F + \sqrt{\sum_{i=r+1}^{\min\{m,n\}} (\sigma_i(X_0) - \sigma_i(X^\star))^2} \\
&\overset{(b)}{\leq} \|X^\star - X_r^\star\|_F + \|X_0 - X^\star\|_F ,
\end{aligned}
$$

where (a) follows from the triangle inequality, and (b) is due to Mirsky's theorem [77]. Plugging this bound into (47), we get

$$
\left\| U_0 V_0^\top - X_r^\star \right\|_F \leq 2 \|X_0 - X^\star\|_F + 2 \|X^\star - X_r^\star\|_F . \qquad (48)
$$

Now we bound the first term of (48). We have

$$\|X_0\|_F = \frac{1}{L}\|\nabla f(0)\|_F = \frac{1}{L}\|\nabla f(0) - \nabla f(X^\star)\|_F \overset{(a)}{\leq} \|0 - X^\star\|_F = \|X^\star\|_F\,,$$
$$L\,\langle X_0, X^\star\rangle = -\langle \nabla f(0), X^\star\rangle \overset{(b)}{\geq} f(0) - f(X^\star) + \frac{\mu}{2}\|X^\star\|_F^2 \overset{(c)}{\geq} \mu\,\|X^\star\|_F^2\,,$$

where (a) follows from the L-smoothness, and (b) and (c) follow from the μ-strong convexity. Then it follows that

$$\|X_0 - X^\star\|_F^2 = \|X_0\|_F^2 + \|X^\star\|_F^2 - 2\,\langle X_0, X^\star\rangle \leq 2\left(1 - \frac{\mu}{L}\right)\|X^\star\|_F^2\,.$$

Applying this inequality to (48), we get the desired inequality. ■

Acknowledgment. We would like to acknowledge the assistance of volunteers in putting together this example manuscript and supplement.

REFERENCES

[1] S. Aaronson, *The learnability of quantum states*, Proc. A, 463 (2007), pp. 3089–3114.

[2] A. Agarwal, S. Negahban, and M. Wainwright, *Fast global convergence rates of gradient methods for high-dimensional statistical recovery*, in Advances in Neural Information Processing Systems, 2010, pp. 37–45.

[3] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma, *Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages*, in Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 13–24.

[4] A. Anandkumar and R. Ge, *Efficient approaches for escaping higher order saddle points in non-convex optimization*, in Proceedings of the Conference on Learning Theory, 2016, pp. 81–102.

[5] H. Andrews and C. Patterson III, *Singular value decomposition (SVD) image coding*, IEEE Trans. Commun., 24 (1976), pp. 425–432.

[6] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput., 27 (2005), pp. 19–42.

[7] S. Balakrishnan, M. Wainwright, and B. Yu, *Statistical guarantees for the EM algorithm: From population to sample-based analysis*, Ann. Statist., 45 (2017), pp. 77–120.

[8] L. Balzano, R. Nowak, and B. Recht, *Online identification and tracking of subspaces from highly incomplete information*, in Proceedings of the (Allerton), 48th Annual Allerton Conference on Communication, Control, and Computing IEEE, 2010, pp. 704–711.

[9] S. Becker, J. Bobin, and E. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39.

[10] S. Becker, E. Candès, and M. Grant, *Templates for convex cone problems with applications to sparse signal recovery*, Math. Program. Comput., 3 (2011), pp. 165–218.

[11] S. Becker, V. Cevher, and A. Kyrillidis, *Randomized low-memory singular value projection*, in Proceedings of the 10th International Conference on Sampling Theory and Applications, 2013.

[12] J. Bennett and S. Lanning, *The Netflix prize*, in Proceedings of the KDD Cup and Workshop, 2007, p. 35.

[13] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, *Sparse local embeddings for extreme multi-label classification*, in Advances in Neural Information Processing Systems, 2015, pp. 730–738.

[14] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi, *Dropping convexity for faster semi-definite optimization*, in Proceedings of the Conference on Learning Theory, 2016, pp. 530–582.

[15] S. Bhojanapalli, B. Neyshabur, and N. Srebro, *Global optimality of local search for low rank matrix recovery*, in Advances in Neural Information Processing Systems, 2016, pp. 3873–3881.

[16] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Trans. Automation Sci. Engrg., 3 (2006), pp. 360–371.

[17] N. Boumal, *Optimization and Estimation on Manifolds*, Ph.D. thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2014.
[18] N. Boumal and P.-A. Absil, *RTRMC: A Riemannian trust-region method for low-rank matrix completion*, in Advances in Neural Information Processing Systems, 2011, pp. 406–414.
[19] N. Boumal, V. Voroninski, and A. Bandeira, *The non-convex Burer-Monteiro approach works on smooth semidefinite programs*, in Advances in Neural Information Processing Systems, 2016, pp. 2757–2765.
[20] N. Boumal, *A Riemannian Low-Rank Method for Optimization over Semidefinite Matrices with Block-Diagonal Constraints*, preprint, arXiv:1506.00575, 2015.
[21] S. Burer and R. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357.
[22] S. Burer and R. Monteiro, *Local minima and convergence in low-rank semidefinite programming*, Math. Program., 103 (2005), pp. 427–444.
[23] J. Cai, E. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.
[24] E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski, *Phase retrieval via matrix completion*, SIAM Rev., 57 (2015), pp. 225–251.
[25] E. Candes and Y. Plan, *Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements*, IEEE Trans. Inform. Theory, 57 (2011), pp. 2342–2359.
[26] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.
[27] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos, *Supervised learning of semantic classes for image annotation and retrieval*, IEEE Trans. Pattern Anal. Machine Intell., 29 (2007), pp. 394–410.
[28] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward, *Coherent matrix completion*, in Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 674–682.
[29] Y. Chen and S. Sanghavi, *A general framework for high-dimensional estimation in the presence of incoherence*, in Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing, IEEE, 2010, pp. 1570–1576.
[30] Y. Chen and M. Wainwright, *Fast Low-Rank Estimation by Projected Gradient Descent: General Statistical and Algorithmic Guarantees*, preprint, arXiv:1509.03025, 2015.
[31] K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. Dhillon, and A. Tewari, *Prediction and clustering in signed networks: A local to global perspective*, J. Mach. Learn. Res., 15 (2014), pp. 1177–1213.
[32] M. Cohen, J. Nelson, and D. Woodruff, *Optimal approximate matrix product in terms of stable rank*, in LIPIcs. Leibniz Int. Proc. Inform. 55, Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2016.
[33] M. Collins, S. Dasgupta, and R. Schapire, *A generalization of principal components analysis to the exponential family*, in Advances in Neural Information Processing Systems, 2001, pp. 617–624.
[34] M. Davenport, Y. Plan, E. van den Berg, and M. Wootters, 1-*bit matrix completion*, Inform. Inference, 3 (2014), pp. 189–223.
[35] D. DeCoste, *Collaborative prediction using ensembles of maximum margin matrix factorizations*, in Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 249–256.
[36] P. Drineas and R. Kannan, *Fast Monte-Carlo algorithms for approximate matrix multiplication*, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, 2001, pp. 452–459.
[37] P. Drineas, R. Kannan, and M. Mahoney, *Fast Monte Carlo algorithms for matrices* I: *Approximating matrix multiplication*, SIAM J. Comput., 36 (2006), pp. 132–157.
[38] A. Edelman, T. Arias, and S. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
[39] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, 2002.
[40] M. Fazel, E. Candes, B. Recht, and P. Parrilo, *Compressed sensing and robust recovery of low rank matrices*, in 42nd Asilomar Conference on Signals, Systems and Computers, IEEE, 2008, pp. 1043–1047.
[41] M. Fazel, H. Hindi, and S. Boyd, *Rank minimization and applications in system theory*, in Proceedings of the American Control Conference, Vol. 4, IEEE, 2004, pp. 3273–3278.

[42] R. Ge, J. Lee, and T. Ma, *Matrix completion has no spurious local minimum*, in Advances in Neural Information Processing Systems, 2016, pp. 2973–2981.
[43] D. Gross, Y.-K. Liu, S. Flammia, S. Becker, and J. Eisert, *Quantum state tomography via compressed sensing*, Phys. Rev. Lett., 105 (2010), 150401.
[44] N. Gupta and S. Singh, *Collectively Embedding Multi-Relational Data for Predicting User Preferences*, preprint, arXiv:1504.06165, 2015.
[45] B. Haeffele and R. Vidal, *Global Optimality in Tensor Factorization, Deep Learning, and Beyond*, preprint, arXiv:1506.07540, 2015.
[46] B. Haeffele and R. Vidal, *Structured Low-Rank Matrix Factorization: Global Optimality, Algorithms, and Applications*, preprint, arXiv:1708.07850, 2017.
[47] N. Halko, P.-G. Martinsson, and J. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[48] M. Hardt and M. Wootters, *Fast matrix completion without the condition number*, in Proceedings of the 27th Conference on Learning Theory, 2014, pp. 638–678.
[49] E. Hazan, *Sparse approximate solutions to semidefinite programs*, in Proceedings of LATIN 2008: Theoretical Informatics, Springer, New York, 2008, pp. 306–316.
[50] U. Helmke and J. Moore, *Optimization and Dynamical Systems*, Springer, New York, 2012.
[51] M. Jaggi and M. Sulovsk, *A simple algorithm for nuclear norm regularized problems*, in Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 471–478.
[52] P. Jain, C. Jin, S. Kakade, and P. Netrapalli, *Computing Matrix Square Root via Non Convex Local Search*, preprint, arXiv:1507.05854, 2015.
[53] P. Jain, R. Meka, and I. Dhillon, *Guaranteed rank minimization via singular value projection*, in Advances in Neural Information Processing Systems, 2010, pp. 937–945.
[54] P. Jain, P. Netrapalli, and S. Sanghavi, *Low-rank matrix completion using alternating minimization*, in Proceedings of the 45th Annual ACM Symposium on Theory of Computing, 2013, pp. 665–674.
[55] C. Jin, S. Kakade, and P. Netrapalli, *Provable efficient online matrix completion via nonconvex stochastic gradient descent*, in Advances in Neural Information Processing Systems, 2016, pp. 4520–4528.
[56] C. Johnson, *Logistic matrix factorization for implicit feedback data*, in Advances in Neural Information Processing Systems, Vol. 27, 2014.
[57] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM J. Optim., 20 (2010), pp. 2327–2351.
[58] A. Kalev, R. Kosut, and I. Deutsch, *Quantum tomography protocols with positivity are compressed sensing protocols*, Quantum Inform., 1 (2015), 15018.
[59] R. Keshavan, *Efficient Algorithms for Collaborative Filtering*, Ph.D. thesis, Stanford University, 2012.
[60] R. Keshavan, A. Montanari, and S. Oh, *Matrix completion from a few entries*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2980–2998.
[61] R. H. Keshavan and S. Oh, *A Gradient Descent Algorithm on the Grassman Manifold for Matrix Completion*, preprint, arXiv:0910.5260, 2009.
[62] V. Khrulkov and I. Oseledets, *Desingularization of bounded-rank matrix sets*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 451–471.
[63] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.
[64] D. Kressner, M. Steinlechner, and B. Vandereycken, *Low-rank tensor completion by Riemannian optimization*, BIT, 54 (2014), pp. 447–468.
[65] A. Kyrillidis and V. Cevher, *Matrix recipes for hard thresholding methods*, J. Math. Imaging Vision, 48 (2014), pp. 235–265.
[66] A. Kyrillidis, M. Vlachos, and A. Zouzias, *Approximate matrix multiplication with application to linear embeddings*, in Proceedings of the IEEE International Symposium on Information Theory, 2014, pp. 2182–2186.
[67] R. Larsen, *PROPACK-Software for Large and Sparse SVD Calculations*, http://sun.stanford.edu/rmunk/PROPACK (2004).

[68] S. Laue, *A hybrid algorithm for convex semidefinite optimization*, in Proceedings of the 29th International Conference on International Conference on Machine Learning, Omnipress, 2012, pp. 1083–1090.
[69] K. Lee and Y. Bresler, *ADMiRA: Atomic decomposition for minimum rank approximation*, IEEE Trans. Information Theory, 56 (2010), pp. 4402–4416.
[70] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software Environ. Tools 6, SIAM, Philadelphia, 1998.
[71] X. Li, Z. Wang, J. Lu, R. Arora, J. Haupt, H. Liu, and T. Zhao, *Symmetry, Saddle Points, and Global Geometry of Nonconvex Matrix Factorization*, preprint, arXiv:1612.09296, 2016.
[72] Z. Lin, M. Chen, and Y. Ma, *The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices*, preprint, arXiv:1009.5055, 2010.
[73] X. Liu, Z. Wen, and Y. Zhang, *Limited memory block Krylov subspace optimization for computing dominant singular value decompositions*, SIAM J. Sci. Comput., 35 (2013), pp. A1641–A1668.
[74] Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li, *Neighborhood regularized logistic matrix factorization for drug-target interaction prediction*, PLoS Comput. Biol., 12 (2016), e1004760.
[75] F. Lu, S. Keles, S. Wright, and G. Wahba, *Framework for kernel regularization with application to protein clustering*, Proc. Nat. Acad. Sci. U.S.A., 102 (2005), pp. 12332–12337.
[76] A. Makadia, V. Pavlovic, and S. Kumar, *A new baseline for image annotation*, in Proceedings of Computer Vision—ECCV 2008, Springer, Berlin, 2008, pp. 316–329.
[77] L. Mirsky, *Symmetric gage functions and unitarily invariant norms*, Quart. J. Math., 11 (1960), pp. 50–59.
[78] B. Mishra, A. Apuroop, and R. Sepulchre, *A Riemannian Geometry for Low-Rank Matrix Completion*, preprint, arXiv:1211.1550, 2012.
[79] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, *Fixed-rank matrix factorizations and Riemannian low-rank optimization*, Comput. Statist., 29 (2014), pp. 591–621.
[80] B. Mishra, G. Meyer, and R. Sepulchre, *Low-rank optimization for distance matrix completion*, in 50th IEEE conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 4455–4460.
[81] S. Negahban and M. Wainwright, *Restricted strong convexity and weighted matrix completion: Optimal bounds with noise*, J. Mach. Learn. Res., 13 (2012), pp. 1665–1697.
[82] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, Berlin, 2004.
[83] D. Park, A. Kyrillidis, S. Bhojanapalli, C. Caramanis, and S. Sanghavi, *Provable Burer-Monteiro Factorization for a Class of Norm-Constrained Matrix Problems*, preprint, arXiv:1606.01316, 2016.
[84] D. Park, A. Kyrillidis, C. Carmanis, and S. Sanghavi, *Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017, pp. 65–74.
[85] B. Recht, M. Fazel, and P. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.
[86] J. Rennie and N. Srebro, *Fast maximum margin matrix factorization for collaborative prediction*, in Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 713–719.
[87] A. I. Schein, L. K. Saul, and L. H. Ungar, *A generalized linear model for principal component analysis of binary data*, in Proceedings of AISTATS, 2003.
[88] N. Srebro, J. Rennie, and T. Jaakkola, *Maximum-margin matrix factorization*, in Advances in Neural Information Processing Systems, 2004, pp. 1329–1336.
[89] R. Sun and Z.-Q. Luo, *Guaranteed matrix completion via nonconvex factorization*, in Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science, 2015, pp. 270–289.
[90] J. Tanner and K. Wei, *Normalized iterative hard thresholding for matrix completion*, SIAM J. Sci. Comput., 35 (2013), pp. S104–S125.
[91] M. Tipping and C. Bishop, *Probabilistic principal component analysis*, J. R. Stat. Soc. Ser. B Stat. Methodol., 61 (1999), pp. 611–622.

[92] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, *Low-rank solutions of linear matrix equations via procrustes flow*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning, Vol. 48, 2016, pp. 964–973.

[93] A. Uschmajew and B. Vandereycken, *Greedy rank updates combined with Riemannian descent methods for low-rank optimization*, in Proceedings of the 12th International Conference on Sampling Theory and Applications, 2015.

[94] K. Verstrepen, *Collaborative Filtering with Binary, Positive-Only Data*, Ph.D. thesis, University of Antwerpen, 2015.

[95] I. Waldspurger, A. d'Aspremont, and S. Mallat, *Phase recovery, MaxCut and complex semidefinite programming*, Math. Program., 149 (2015), pp. 47–81.

[96] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang, *Multi-label sparse coding for automatic image annotation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1643–1650.

[97] L. Wang, X. Zhang, and Q. Gu, *A Unified Computational and Statistical Framework for Nonconvex Low-Rank Matrix Estimation*, preprint, arXiv:1610.05275, 2016.

[98] L. Wang, X. Zhang, and Q. Gu, *A unified variance reduction-based framework for nonconvex low-rank matrix recovery*, in Proceedings of the International Conference on Machine Learning, 2017, pp. 3712–3721.

[99] A. Waters, A. Sankaranarayanan, and R. Baraniuk, *SpaRCS: Recovering low-rank and sparse matrices from compressive measurements*, in Advances in Neural Information Processing Systems, 2011, pp. 1089–1097.

[100] K. Weinberger, F. Sha, Q. Zhu, and L. Saul, *Graph Laplacian regularization for large-scale semidefinite programming*, in Advances in Neural Information Processing Systems, 2007, pp. 1489–1496.

[101] J. Weston, S. Bengio, and N. Usunier, *WSABIE: Scaling up to large vocabulary image annotation*, in Proceedings of IJCAI, 2011, pp. 2764–2770.

[102] M. Wootters, *private communication*, 2016.

[103] X. Yi, D. Park, Y. Chen, and C. Caramanis, *Fast algorithms for robust PCA via gradient descent*, in Advances in Neural Information Processing Systems, 2016, pp. 4152–4160.

[104] A. Yurtsever, Q. Tran-Dinh, and V. Cevher, *A universal primal-dual convex optimization framework*, in Advances in Neural Information Processing Systems, Vol. 28, 2015, pp. 3132–3140.

[105] D. Zhang and L. Balzano, *Global Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation*, preprint, arXiv:1506.07405, 2015.

[106] T. Zhao, Z. Wang, and H. Liu, *A nonconvex optimization framework for low rank matrix estimation*, in Advances in Neural Information Processing Systems, Vol. 28, 2015, pp. 559–567.

[107] Q. Zheng and J. Lafferty, *A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements*, in Advances in Neural Information Processing Systems, 2015, pp. 109–117.

[108] Q. Zheng and J. Lafferty, *Convergence Analysis for Rectangular Matrix Completion Using Burer-Monteiro Factorization and Gradient Descent*, preprint, arXiv:1605.07051, 2016.

[109] G. Zhou, W. Huang, K. Gallivan, P. Van Dooren, and P.-A. Absil, *A Riemannian rank-adaptive method for low-rank optimization*, Neurocomputing, 192 (2016), pp. 72–80.

[110] Z. Zhu, Q. Li, G. Tang, and M. Wakin, *The Global Optimization Geometry of Nonsymmetric Matrix Factorization and Sensing*, preprint, arXiv:1703.01256, 2017.