A Non-Euclidean Gradient Descent Framework for Non-Convex Matrix Factorization

Ya-Ping Hsieh, Yu-Chun Kao, Rabeeh Karimi Mahabadi, Alp Yurtsever, Anastasios Kyrillidis, *Member, IEEE,* and Volkan Cevher, *Senior Member, IEEE*

***Abstract*—We study convex optimization problems that feature low-rank matrix solutions. In such scenarios, non-convex methods offer significant advantages over convex methods due to their lower space complexity, as well as practical faster convergence. Under mild assumptions, these methods feature global convergence guarantees.**

In this paper, we extend the results on this matter by following a different path. We derive a non-Euclidean optimization framework in the non-convex setting that takes nonlinear gradient steps on the factors. Our framework enables the possibility to further exploit the underlying problem structures, such as sparsity or low-rankness on the factorized domain, or better dimensional dependence of the smoothness parameters of the objectives. We prove that the non-Euclidean methods enjoy the same rigorous guarantees as their Euclidean counterparts, under appropriate assumptions. Numerical evidence with Fourier Ptychography and `FastText` applications, using real data, shows that our approach can enhance solution quality, as well as convergence speed over the standard non-convex approaches.

***Index Terms*—Non-convex optimization, low-rank approximation, non-Euclidean gradient descent**

I. Introduction

We study convex minimization problems with respect to a matrix variable:

$$\min_{X\in\mathcal{X}\subseteq\mathbb{R}^{p\times q}} f(X), \tag{I.1}$$

where $\mathcal{X}$ is either the positive semi-definite (PSD) cone (in which case $p=q$) or the whole space. Let $X^\star$ be the optimal solution of (I.1). We are interested in the scenario where $r^\star \triangleq \text{rank}(X^\star) \ll \min\{p,q\}$. Such a formulation spans a wide spectrum of applications in machine learning and signal processing [1]–[16].

Given low-rankness at the optimum, recent research has suggested the following recipe for solving (I.1): Fix a number $r \ll p, q$ as close as possible to $r^\star$, and factorize $X = UV^\top$ (or $X = UU^\top$ in the PSD case), where $U \in \mathbb{R}^{p\times r}$ and $V \in \mathbb{R}^{q\times r}$. Then, recast (I.1) as:

$$\min_{U,V}\ g(U,V) \coloneqq f(UV^\top). \tag{I.2}$$

Since the program (I.2) is non-convex, it is impossible to prove global convergence without additional assumptions. To this end, the typical approach is to assume that the initialization is close to the global optimum in some sense, and prove that simple gradient descent for U and V provably converges to the *global* minimum.

In theory, such an approach relies on unverifiable initialization conditions and hence is not fully satisfactory. Nonetheless, it has yielded wide success in practice [5], [13], [17], [18]. In particular, the assumption of good initialization can often be met using heuristics, for instance multiple trials of random initialization or running a few iterations of gradient descent on the original matrix variable space; see, for instance, Section VI.

Following the recipe (I.2), we ask whether we can further exploit the problem structures in the non-convex setting. For instance, in phase retrieval, the decision variable $X \in \mathbb{R}^{d^2\times d^2}$ of the convex problem (I.1) is obtained by lifting a vectorized image $U \in \mathbb{R}^{d^2\times 1}$. However, the original image, whose natural domain is $\mathbb{R}^{d\times d}$, often exhibits further low-rankness, a useful structure not revealed in the vectorized form. Is there an algorithm that directly runs in $\mathbb{R}^{d\times d}$ and features low-rank updates, similar to the Frank-Wolfe method [19] in the convex case, while retaining the guarantees enjoyed within the non-convex research vein?

As another motivating example, recent studies in computer science [20] and machine learning [21]–[23] have shown that the log-softmax function [24], with important applications in deep learning and natural language processing, converges much faster when a nonlinear operation is applied to each gradient step. Is there an analogous result in the non-convex setting, again retaining the favorable global convergence?

In this paper, we show that the above-posed questions can be addressed by the *non-Euclidean optimization framework* in a unified fashion. To promote sparse (or low-rank) iterates, such as for the phase retrieval application, we show that gradient descent on U in the *nuclear* norm enjoys rank-1 updates in the natural image domain $\mathbb{R}^{d\times d}$. For optimizing the log-softmax function, we employ gradient descent in the *spectral* norm, and show that the advantages observed in [20]–[23] carry over to the non-convex setting.

Most importantly, we prove that, *under the similar assumption of a good initialization, our non-Euclidean methods provably converge to the global optimum*.

Akin to previous work, we empirically verify the initialization assumption, through extensive experiments on the *real* data. Numerical evidence with Fourier Ptychography and `FastText` applications shows that our approach can significantly enhance solution quality as well as speed over the standard non-convex approaches.

Related work: For solving (I.1) with PSD constraint, [25] and [26] popularized the factorization idea leading to the formulation (I.2). In recent years, there has been a large body of literature [13], [27]–[32] studying the convergence guarantees under factorization, while most of them only apply to the

quadratic loss. For generic convex loss, [17] focused on (I.1) with PSD constraint. The analysis was further extended in [18] to unconstrained problems. Another recent work [33] studied the convergence from both statistical and algorithmic perspectives, with distinct assumptions.

To the best of our knowledge, we are the first to introduce and analyze non-Euclidean gradient steps for solving the factorized formulation (I.2). All of the aforementioned works employ Euclidean gradient descent steps for variables U, V in (I.2).

On a related note, the work [21] studied the spectral gradient method for optimizing $\log$-softmax functions in the deep learning realm, which reduces to matrix factorization when a two-layered neural network is considered. However, no convergence guarantee to the *global* optimum was provided.

For completeness, we further mention another line of research which focuses on the Riemannian geometry of matrices; see [34] for a comprehensive survey and [35], [36] for recent developments. Despite also having a non-Euclidean gradient feature, these works are distinct from our work as they do not utilize the factorization in (I.2). The only exception we know of is [37], where the convergence guarantees for Riemannian first-order methods are proved for (I.1) with linear objectives under PSD + affine constraints. It is interesting to see if the techniques in [37] can be used to analyze general f under our setting, or whether our non-Euclidean algorithms succeed for the tasks in [37].

II. Background

A. Notations

Given a matrix X, we use $\sigma_i(X)$ to denote its i-th largest singular value. We use $\|\cdot\|_*$, $\|\cdot\|_{S_\infty}$ and $\|\cdot\|_F$ to denote nuclear norm, spectral norm and Frobenius norm, respectively. The Schatten-p norm of a matrix X, denoted by $||X||_{S_p}$, is defined as $\left(\sum_i \sigma_i^p(X)\right)^{1/p}$.

We define a parameter $\tau(X) \equiv \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)}$. X_r denotes the best rank-r approximation of X, and therefore $\sigma_i(X_r) = \sigma_i(X)$ for $1 \le i \le r$. For a given matrix U, we use Q_U to denote the matrix constituted of an orthonormal basis of the column space of U. Note that $Q_U Q_U^T$ is the projection operator of the column space of U and thus $Q_U Q_U^T U = U$. Given two matrices $X, Y \in \mathbb{R}^{p\times q}$, the Hilbert-Schmidt inner product is denoted by $\langle X, Y\rangle = \mathrm{Tr}(X^\top Y)$.

For two real numbers a and b, the minimum of them is denoted by $a \wedge b$.

B. Matrix operators

For any matrix X, let $X = P\Lambda R^\top$ be its singular value decomposition (SVD). We define: leftmargin=0.5cm

- The *nuclear #-operator*: Let $P_{\max}$ and $R_{\max}$ be the left and right singular vectors corresponding to the largest singular value of X. Then, the *nuclear #-operator* corresponds to

$$[X]_*^{\#} \triangleq \sigma_1(X) P_{\max} R_{\max}^\top. \tag{II.1}$$

  That is, $[X]_*^{\#}$ is the best rank-1 approximation of X.

- The *spectral #-operator*: Let $\mathrm{rank}(X) = r$. Then, the *spectral #-operator* corresponds to

$$[X]_\infty^{\#} \triangleq \left(\sum_i \sigma_i(X)\right) \cdot P I_r R^\top = \|X\|_* \cdot P I_r R^\top, \tag{II.2}$$

  where $I_r \in \mathbb{R}^{p\times q}$ has 1's on the first r diagonal entries, and 0 otherwise. Notice that $[X]_\infty^{\#}$ has the same rank as X, but with singular values all equal to $\|X\|_*$.

To motivate the above definition and notation, we remark that (II.1) and (II.2) are instances of the so-called *duality map* in Banach spaces [38]. Let $(\mathcal{X}^\star, \|\cdot\|^\star)$ be a general Banach space[1], $(\mathcal{X}, \|\cdot\|)$ be its dual space, and let $\langle\cdot,\cdot\rangle : \mathcal{X}^\star \times \mathcal{X} \to \mathbb{R}$ denote the dual pair [39]. The (possibly set-valued) duality map $\# : \mathcal{X}^\star \to \mathcal{X}$ maps a point $X \in \mathcal{X}^\star$ to an element of the dual space $X^\# \in \mathcal{X}$ satisfying the following relation:

$$\left\langle X, X^\# \right\rangle = \|X^\#\|^2 = \left(\|X\|^\star\right)^2. \tag{II.3}$$

One can easily verify that $[\cdot]_*^{\#}$ is the duality map when $(\mathcal{X}, \|\cdot\|) = (\mathbb{R}^{p\times q}, \|\cdot\|_*)$; that is, the following relation holds for any $X \in \mathbb{R}^{p\times q}$:

$$\left\langle X, [X]_*^{\#} \right\rangle = \|[X]_*^{\#}\|_*^2 = \|X\|_{S_\infty}^2, \tag{II.4}$$

in which case the dual pair becomes the Hilbert-Schmidt inner product. Similarly, $[\cdot]_\infty^{\#}$ is the duality map when $(\mathcal{X}, \|\cdot\|) = (\mathbb{R}^{p\times q}, \|\cdot\|_\infty)$.

We quote some properties of the nuclear and spectral #-operators.

Properties 1. *For any differentiable function f, it holds*

$$(\forall X, Y) \quad \|\nabla f(Y) - \nabla f(X)\|_* \le L\|Y - X\|_{S_\infty} \tag{II.5}$$

if and only if

$$(\forall X, Y) \quad f(Y) \le f(X) + \langle \nabla f(X), Y - X\rangle + \frac{L}{2}\|Y - X\|_{S_\infty}^2. \tag{II.6}$$

Moreover,

$$X - \frac{1}{L}[\nabla f(X)]_\infty^{\#} \in \arg\min_Y f(X) + \langle \nabla f(X), Y - X\rangle + \frac{L}{2}\|Y - X\|_{S_\infty}^2. \tag{II.7}$$

Also, for any differentiable function f, it holds

$$(\forall X, Y) \quad \|\nabla f(Y) - \nabla f(X)\|_{S_\infty} \le L\|Y - X\|_* \tag{II.8}$$

if and only if

$$(\forall X, Y) \quad f(Y) \le f(X) + \langle \nabla f(X), Y - X\rangle + \frac{L}{2}\|Y - X\|_*^2. \tag{II.9}$$

Moreover,

$$X - \frac{1}{L}[\nabla f(X)]_*^{\#} \in \arg\min_Y f(X) + \langle \nabla f(X), Y - X\rangle + \frac{L}{2}\|Y - X\|_*^2. \tag{II.10}$$

[1]The reason why we choose $\mathcal{X}^\star$, instead of the normal $\mathcal{X}$, to denote the underlying space is due to the fact that our algorithms use #-operators on the *gradients*, which naturally live in the dual space.

One can see conditions (II.5), (II.6), (II.8), and (II.9) as similar to the classic smoothness definition, but in different norms. The proof of (II.5) $\equiv$ (II.6) and (II.8) $\equiv$ (II.9) can be found in [20], [40]. A simple derivation of (II.7) can be found in [21]; (II.10) is proved using similar techniques.

C. Reivew of Euclidean Methods

Recall the objective in (I.1) and the factorized form in (I.2). We first focus on the PSD-constrained case

$$\min_{U\in\mathbb{R}^{p\times r}} g(U) \triangleq \min_{U\in\mathbb{R}^{p\times r}} f(UU^\top). \tag{II.11}$$

The simplest algorithm for solving (II.11) is to do gradient descent on U, which corresponds to the iterates

$$U_{i+1} = U_i - \eta_i \nabla g(U_i) = U_i - \eta_i \nabla f(U_i U_i^\top)\cdot U_i. \tag{II.12}$$

We adopt the same settings as [17], [18], who analyzed (II.12) applied to (II.11). The high-level message of these work is that, as alluded to in the introduction, a good initilization is sufficient to ensure convergence to the global optimum; cf., **Theorem 4.1** of [17] and **Theorem 4.4** of [18].

III. Non-Convex Nuclear Gradient Methods for PSD-Constrained Problems

We consider the *Nuclear Gradient Descent* for solving (II.11). Before giving convergence guarantees, let us first motivate with a concrete example.

Consider the phase retrieval application[2]:

$$\min_{x\in\mathbb{R}^{d\times d}} \sum_{i=1}^{n} \left(b_i - |\langle a_i, x\rangle|^2\right)^2 \tag{III.1}$$

where $b_i = |\langle a_i, x^\natural\rangle|^2 + w_i$ is the noisy observation under the true image $x^\natural \in \mathbb{R}^{d\times d}$, the measurement $a_i \in \mathbb{R}^{d\times d}$, and noise w_i. The inner product here is in the Hilbert-Schmidt sense.

As (III.1) is a non-convex problem, the standard approach first vectorizes $x^\natural$ and a_i into $U^\natural \in \mathbb{R}^{d^2\times 1}$ and $A_i \in \mathbb{R}^{d^2\times 1}$, and then rewrites the observations in the equivalent form:

$$b_i = \mathrm{Tr} A_i A_i^\top U^\natural U^{\natural\top} + w_i. \tag{III.2}$$

Renaming $\mathcal{A}_i \coloneqq A_i A_i^\top$ and $X^\natural \coloneqq U^\natural U^{\natural\top}$, the program (III.1) now turns into a convex problem

$$\min_{X\in\mathbb{R}^{d^2\times d^2}} \|b - \mathcal{A}(X)\|_2^2 \tag{III.3}$$

for an appropriate linear operator $\mathcal{A}$. The solution we wish to recover, the $X^\natural$, is then rank-1.

The program (III.3) is of the form (II.11), and hence existing gradient methods apply. However, the non-convex framework in [17] sets $r = 1$ (or a small number $r \ll d$) in (II.11), and hence the image is now viewed as a vector in $\mathbb{R}^{d^2\times 1}$. Such an operation does not utilize the underlying structure of natural images, which exhibit low-rankness when viewed as in $\mathbb{R}^{d\times d}$.

In this section, we show that the non-Euclidean methods provide a framework for *simultaneously* exploiting the computational efficiency of factorized gradient methods, and the additional low-rank structures of natural images. We achieve the desiderata in two steps. First, we show that the **nuclear** gradient method, for any factorization, gives rise to rank-1 updates, and we prove that the nuclear gradient method possesses similar convergence guarantees to the Euclidean counterpart. Second, we consider a general factorization through **tensor product**, which allows us to preserve the structure of images even in the factorized domain U. Finally, if the objective is strongly convex, we provide a variant of our algorithm achieving linear rate.

[2]Strictly speaking, the variable x in (III.1) should be $\mathbb{C}^{d\times d}$ or $\mathbb{R}^{2d\times 2d}$. We write $x \in \mathbb{R}^{d\times d}$ for notational convenience. Same for the measurements a_i's.

A. Convergence Rate of Nuclear Gradient Descent for PSD-Constrained Programs

We propose Algorithm 1, which is obtained by simply applying the $[\cdot]_*^\#$-operator to the gradients in (II.12).

Algorithm 1 Nuclear Gradient Descent for (II.11)

Input: $X_0 = U_0 U_0^\top$, step-sizes η_i.
for $i = 0, 1, \ldots, k-1$ **do**
  $U_{i+1} = U_i - \eta_i [\nabla f(U_i U_i^\top)\cdot U_i]_*^\#$
end for
Return: U_k

Let $X^\star = U^\star (U^\star)^\top$ be the global optimum, and define

$$D_*(U_1, U_2) \equiv \min_{R \text{ is unitary}} \|U_1 - U_2 R\|_*. \tag{III.4}$$

Under a good initialization, we prove that Algorithm 1 converges to the global optimum.

Theorem 1. *Assume that rank$(X^\star) = r$, $\nabla f(\cdot) \in \mathbb{R}^{p\times p}$ is symmetric and f being convex and $L_{S_1\to S_\infty}$-smooth: $\|\nabla f(X) - \nabla f(Y)\|_{S_\infty} \le L_{S_1\to S_\infty}\|Y - X\|_*$. Assume also that*

$$\tilde{D}_* \equiv \max_{U: f(UU^\top)\le f(U_0U_0^\top)} D_*(U, U^\star) \le \frac{\sigma_r(U^\star)}{10}. \tag{III.5}$$

If the step-size is chosen according to $\eta_i \le \gamma_i \equiv \frac{1}{4}\left(\frac{1}{L_{S_1\to S_\infty}\|X_i\|_{S_\infty}} \wedge \frac{1}{\|\nabla f(X_i)\|_{S_\infty}}\right)$, then we have

$$f(U_k U_k^\top) - f(U^\star U^{\star\top}) \le \frac{4.5\,\tilde{D}_*^2}{\sum_{i=0}^{k-1}\eta_i},$$

and $\min_i \gamma_i \ge \frac{1}{4}\overline{\eta}$, where

$$\overline{\eta} \equiv \Big(\frac{1}{L_{S_1\to S_\infty}(\frac{11}{9})^2\|X_0\|_{S_\infty}} \wedge \frac{1}{\frac{40 L_{S_1\to S_\infty}}{81}\sigma_r(U_0)\sigma_1(U_0) + \|\nabla f(X_0)\|_*}\Big).$$

In particular, to avoid computing γ_i at each iteration, we can simply set $\eta_i = \overline{\eta}$ and still attain the convergence rate $O\left(\frac{1}{k}\right)$.

Proof. See Section A in the supplementary material. □

Remark 1. *It is worth noting that the smoothness assumption is on the original convex objective $f(X)$, not the factorized problem $f(UU^\top)$. The same remark applies to **Theorem 2** and **3** below.*

Algorithm 2 Nuclear Gradient Descent for (III.7)

Input: Initial point $X_0 = U_0 \otimes U_0$, step-sizes η_i.
for $i = 0, 1, \ldots, k-1$ **do**
$U_{i+1} = U_i - \eta_i [\nabla f(U_i \otimes U_i) \cdot U_i]^{\#}_*$
end for
Return: U_k

B. Factorizing through Tensor Products

Let $A \in \mathbb{R}^{d_1 \times d_2}$ and $B \in \mathbb{R}^{d_3 \times d_4}$ be two matrices. Their tensor product $A \otimes B \in \mathbb{R}^{d_1 d_3 \times d_2 d_4}$ is given, in the block matrix form, by

$$A \otimes B = [a_{ij} B]_{ij} \tag{III.6}$$

where a_{ij} is the (i,j)-th entry of A.

Recall the convex formulation of the phase retrieval problem (III.3). Instead of factorizing the variable $X \in \mathbb{R}^{d^2 \times d^2}$ into $UU^\top$ for some $U \in \mathbb{R}^{d^2 \times 1}$, we now consider the factorization through tensor product. That is, we consider the factorized variable as $U \in \mathbb{R}^{d \times d}$, and we decompose the original problem (II.11) as

$$\min_{U \in \mathbb{R}^{d \times d}} g(U) \triangleq f(U \otimes U). \tag{III.7}$$

Evidently, the program (III.7) still preserves the rank-1 property of the solution $X^\star \in \mathbb{R}^{d^2 \times d^2}$ to the convex problem, as we can always vectorize the solution to (III.7) into $U^\star \in \mathbb{R}^{d^2 \times 1}$ and output $X^\star := U^\star U^{\star\top}$. However, notice now that the decision variable operates in $\mathbb{R}^{d \times d}$, which is the natural ambient space for images.

Motivated by the above observations, we propose Algorithm 2, which is the analogue of Algorithm 1 with the tensor product factorization.

The theorem and analysis of **Theorem 1** generalizes immediately to the above algorithm, except that the last term of (III.5) needs to be replaced by the equivalent quantity $\frac{\sqrt{\sigma_r(X^*)}}{10}$. We provide a complete proof in Section B of the supplementary material.

C. Linear Rate for Smooth and Strongly Convex Objectives

In this subsection, we show that linear convergence can be attained for smooth and strongly convex objectives, as in classical convex optimization theory.

We apply Algorithm 3 to solve (II.11). Notice the subtle difference between Algorithm 1 and Algorithm 3: The updates of Algorithm 1 are based on $[\nabla f(UU^\top) \cdot U]^{\#}_*$, whereas Algorithm 3 uses $[\nabla f(UU^\top)]^{\#}_* \cdot U$.

Algorithm 3 Nuclear Gradient Descent for (II.11)

Input: Initial point $X_0 = U_0 U_0^\top$, step-sizes η_i.
for $i = 0, 1, \ldots, k-1$ **do**
$U_{i+1} = U_i - \eta_i [\nabla f(U_i U_i^\top)]^{\#}_* \cdot U_i$
end for
Return: U_k

Let $X^\star$ be the global optimum and denote its best rank-r approximation as $X^\star_r = U^\star (U^\star)^\top$. If $X^\star$ is exactly rank-r, then $X^\star \equiv X^\star_r$.

Theorem 2. *Assume $\nabla f(\cdot) \in \mathbb{R}^{p \times p}$ is symmetric. Let f be both L-smooth $\|\nabla f(X) - \nabla f(Y)\|_{S_\infty} \le L \|Y - X\|_*$ and μ-strongly convex $f(Y) \ge f(X) + \langle \nabla f(X), Y - X \rangle + \frac{\mu}{2}\|Y - X\|^2_*$. Denote $\kappa = \frac{L}{\mu}$ and define*

$$D_F(U_1, U_2) \equiv \min_{R \text{ is unitary}} \|U_1 - U_2 R\|_F, \tag{III.8}$$
$$\tilde{D}_F \equiv \max_{U : f(UU^\top) \le f(U_0 U_0^T)} D_F(U, U^\star),$$
$$\tilde{D}_* \equiv \max_{U : f(UU^\top) \le f(U_0 U_0^T)} D_*(U, U^\star),$$

and $\rho \equiv \frac{1}{100 \kappa \tau(X^\star_r)}$.

Assume that $\tilde{D}_F \le \rho \sigma_r(U^\star_r)$, $\|X^\star - X^\star_r\|_F \le \frac{1}{200 \kappa^{1.5} \tau(X^\star_r)} \sigma_r(X^\star)$, and $\tilde{D}_* \le \frac{1}{81\kappa} \frac{\sigma_r(X^\star)}{\sigma_1(U^\star)}$. If we choose step-sizes as $\eta_i = \frac{1}{16(L\|U_i U_i^\top\|_{S_\infty} + \|\nabla f(U_i U_i^\top)\# Q_{U_i} Q_{U_i}^\top\|_{S_\infty})}$, then we have

$$D_F(U_{i+1}, U^\star_r)^2 \le \alpha_i D_F(U_i, U^\star_r)^2 + \beta_i \|X^\star - X^\star_r\|_F \tag{III.9}$$

where $\alpha_i = 1 - \frac{0.7\, \mu\, \sigma_r(X^\star)}{2} \eta_i$ and $\beta_i = \frac{L}{2}\eta_i$. We also have $\min_{i \ge 0} \eta_i \ge \frac{1}{16}\overline{\eta}$, where

$$\overline{\eta} \equiv \frac{1}{L\left(\frac{1+\rho}{1-\rho}\right)^2 \|X_0\|_{S_\infty} + \frac{4L\sigma_1(U_0)\sigma_r(X^\star)}{81\kappa\sigma_1(U^\star)(1-\rho)} + \|\nabla f(X_0)\|_{S_\infty}}.$$

That is, when the rank of the optimum $X^\star$ is equal to or less than r, then we have linear convergence in the distance measure $D_F(U_k, U^\star) \le \bar{\alpha}^k D_F(U_0, U^\star)$ where $\bar{\alpha} \triangleq 1 - \frac{0.7\, \mu\, \sigma_r(X^\star)}{2}\overline{\eta} < 1$.

Proof. See Section C in the supplementary material. □

The above theorem highlights that our framework applies to approximately low-rank minimizers. Given a minimizer $X^\star$ with $\mathrm{rank}(X^\star) = r^\star$, assume that we have factorized the problem with rank r. Then, the analysis (after replacing $\sigma_r(U_i)$ by $\max\{\sigma_r(U^\star), \sigma_r(U_i)\}$) shows that our algorithms converge to the best rank-r approximation of $X^\star$ if $r < r^\star$, and converge to $X^\star$ if $r > r^\star$, with the same rate.

IV. Non-Convex Spectral Gradient Methods for PSD-Constrained Problems

We consider the *Spectral Gradient Descent* for solving (II.11). Our main motivation is to tackle the matrix version of the log-sum-exp function (IV.1), which has important applications in deep learning and natural language processing.

Consider the log-sum-exp function over vectors $z \in \mathbb{R}^d$:

$$\mathrm{lse}(z) = \log \sum_{i=1}^{d} \exp(z_i), \tag{IV.1}$$

which is obtained by applying Nesterov's smoothing to the $\max$ function [41]. It arises naturally as the main part of the log-softmax function [24] in machine learning.

Standard calculation shows

$$\forall z, z' \in \mathbb{R}^d \quad \|\nabla \mathrm{lse}(z) - \nabla \mathrm{lse}(z')\|_2 \le \frac{1}{2}\|z - z'\|_2, \tag{IV.2}$$

which implies that lse is $\frac{1}{2}$-smooth in the Euclidean norm. Using $\|\cdot\|_2 \le \sqrt{d}\|\cdot\|_\infty$ and $\|\cdot\|_2 \ge \frac{1}{\sqrt{d}}\|\cdot\|_1$, one expects that

Algorithm 4 Spectral Gradient Descent for (II.11)

Input: $X_0 = U_0 U_0^\top$, step-sizes η_i.
for $i = 0, 1, \ldots, k-1$ **do**
$U_{i+1} = U_i - \eta_i [\nabla f(U_i U_i^\top) \cdot U_i]^{\#}_{\infty}$
end for
Return: U_k

lse should be $\frac{d}{2}$-smooth in the ℓ_∞-norm. However, a careful analysis [41] gives

$$\forall z, z' \in \mathbb{R}^d \quad \|\nabla \text{lse}(z) - \nabla \text{lse}(z')\|_1 \le \|z - z'\|_\infty \tag{IV.3}$$

which reveals that lse is in fact 1-smooth in the ℓ_∞-norm, vastly improving upon the naïve estimate $\frac{d}{2}$. In the vector case, the property (IV.3) hints upon the use of the *spectral* $\#$-operator, which has led to impressive progress in computer science [20] and machine learning [21]–[23].

We propose to perform spectral $\#$-operator on the matrix problems, as there are important matrix variants of log-sum-exp function; see the `FastText` application in Section VI. The convergence is analyzed in Section IV-A, and in Section IV-B we show that the same calculation leading to (IV.3) generalizes to the matrix-variate lse as well.

A. Convergence Rate of Spectral Gradient Descent for PSD-Constrained Programs

We consider Algorithm 4, which is obtained by applying the $[\cdot]^{\#}_{\infty}$-operator to the gradient updates.

Let $X^\star = U^\star (U^\star)^\top$ be the global optimum. Define

$$D_\infty(U_1, U_2) \equiv \min_{R \text{ unitary}} \|U_1 - U_2 R\|_{S_\infty}. \tag{IV.4}$$

Similar to **Theorem 1**, under a good initialization, we can guarantee the convergence to the global optimum.

Theorem 3. *Assume that rank$(X^\star) = r$, $\nabla f(\cdot) \in \mathbb{R}^{p \times p}$ is symmetric, and f is convex and $L_{S_\infty \to S_1}$-smooth,* i.e., *$\|\nabla f(X) - \nabla f(Y)\|_* \le L_{S_\infty \to S_1} \|Y - X\|_{S_\infty}$. Assume also that*

$$\tilde{D}_\infty \equiv \max_{U : f(UU^\top) \le f(U_0 U_0^\top)} D_\infty(U, U^\star) \le \frac{\sigma_r(U^\star)}{10}. \tag{IV.5}$$

If the step-size is chosen according to $\eta_i \le \gamma_i \equiv \frac{1}{4}\left(\frac{1}{L_{S_\infty \to S_1}\|X_i\|_{S_\infty}} \wedge \frac{1}{\|\nabla f(X_i)_r\|_*}\right)$, then we have after k iterations:

$$f(U_k U_k^\top) - f(U^\star U^{\star\top}) \le \frac{4.5\ \tilde{D}_\infty^2}{\sum_{i=0}^{k-1} \eta_i}, \tag{IV.6}$$

and $\min_i \gamma_i \ge \frac{1}{4}\overline{\eta}$, where

$$\overline{\eta} \equiv \Big(\frac{1}{L_{S_\infty \to S_1}(\frac{11}{9})^2 \|X_0\|_{S_\infty}} \wedge \frac{1}{\frac{40 L_{S_\infty \to S_1}}{81}\sigma_r(U_0)\sigma_1(U_0) + \|\nabla f(X_0)_r\|_*}\Big).$$

In particular, to avoid computing γ_i at each iteration, we can simply set $\eta_i = \overline{\eta}$ and still attain the convergence rate $O\left(\frac{1}{k}\right)$.

Proof. See Section D in the supplementary material. □

B. Convergence Comparison the Matrix-Variate lse Function

Let f be a matrix-variate function of the form

$$f(A) = \text{lse}(Ax), \quad A \in \mathbb{R}^{d' \times d} \tag{IV.7}$$

where $x \in \mathbb{R}^d$ is a fixed vector and the lse function is given in (IV.1)[3]. Such functions appear, for instance, in the final layer of deep neural networks [24] or the `FastText` application [42], [43].

We show that the smoothness parameters for (IV.7) exhibit similar comparison as (IV.2) and (IV.3) in the vector case.

Lemma 1. *Let $f(A) \coloneqq lse(Ax)$ for a fixed vector x. Then f is convex. Moreover, for all $A, A' \in \mathbb{R}^{d' \times d}$, we have*

$$\|\nabla f(A) - \nabla f(A')\|_F \le \frac{1}{2}\|x\|_2^2 \cdot \|A - A'\|_F \tag{IV.8}$$

and

$$\|\nabla f(A) - \nabla f(A')\|_{S_1} \le \|x\|_2^2 \cdot \|A - A'\|_{S_\infty}. \tag{IV.9}$$

In other words, $L_{S_2 \to S_2} = \frac{\|x\|_2^2}{2}$ and $L_{S_\infty \to S_1} = \|x\|_2^2$.

Proof. See Section E in the supplementary material. □

We now compare the convergence rates between **Algorithm 4** and the Euclidean method with, say, Gaussian initialization, applied to the matrix lse function (IV.7). Without loss of generality, assume that $\|x\|_2^2 = 1$ (otherwise one can define a new decision variable $\tilde{A} \coloneqq \|x\|_2^2 \cdot A$ and minimize over $\tilde{A}$) and $d > d'$. Then the bound (IV.6) dictates the convergence rate

$$f(X_k) - f(X^\star) = O\left(\frac{d}{k}\right),$$

whereas the Euclidean counterpart (see equation (9) in [17]) is

$$f(X_k) - f(X^\star) = O\left(\frac{\sqrt{d'}d}{k}\right).$$

As a result, by exploiting the favorable S_∞ geometry for the matrix lse function, one can obtain an $O(\sqrt{d'})$ improvement over standard gradient method, which can be significant when the dimension is large.

V. Convergence Rate for non-PSD Programs

So far, we have only considered PSD-constrained problems (II.11). In this section, we show that the guarantees in previous sections can be extended to unconstrained programs via a lifting trick [52].

Consider the asymmetrically factorized program:

$$\min_{X \in \mathbb{R}^{p \times q}} f(X) \triangleq \min_{U \in \mathbb{R}^{p \times r}, V \in \mathbb{R}^{q \times r}} f(UV^\top). \tag{V.1}$$

Define $W = \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{(p+q) \times r}$, and define a new objective by

$$\hat{f}(WW^\top) = \hat{f}\left(\begin{bmatrix} UU^\top & UV^\top \\ VU^\top & VV^\top \end{bmatrix}\right) \coloneqq f(UV^\top). \tag{V.2}$$

It is easy to verify the following: leftmargin=0.5cm

[3]We do not assume A to be constrained in the PSD cone in this subsection. The convergence guarantees for general A is given in Section V.

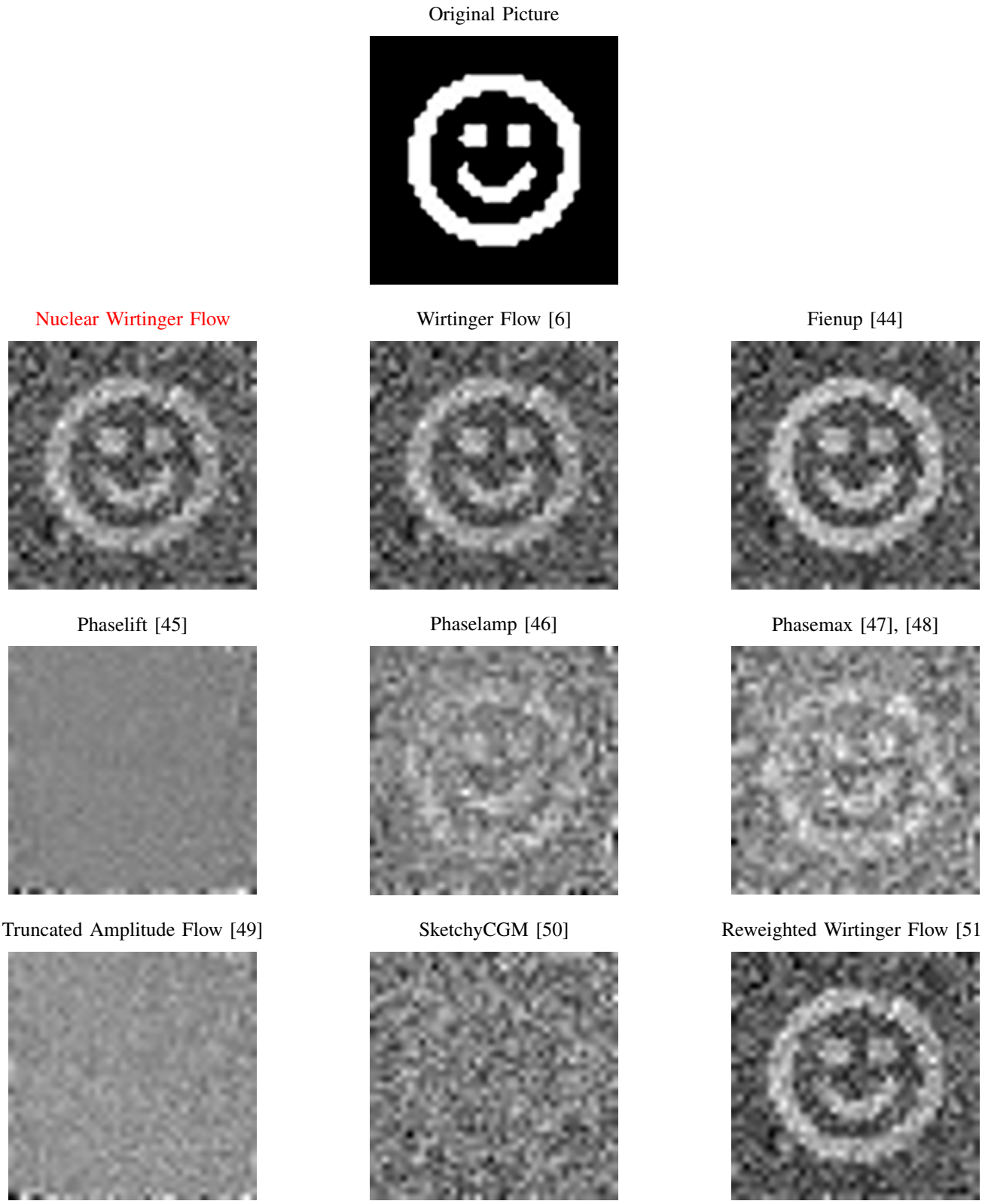


Fig. 1: Comparison of phase retrieval algorithms, synthetic dataset 1.

- $W_{i+1} = W_i - \eta_i[\nabla \hat{f}(W_i W_i^\top)W_i]_*^\#$ is equivalent to

$$\begin{bmatrix} U_{i+1} \\ V_{i+1} \end{bmatrix} = \begin{bmatrix} U_i \\ V_i \end{bmatrix} - \frac{\eta_i}{2} \begin{bmatrix} \nabla f(U_i V_i^\top)V_i \\ \nabla f(U_i V_i^\top)^\top U_i \end{bmatrix}_*^\# . \quad \text{(V.3)}$$

- $W_{i+1} = W_i - \eta_i[\nabla \hat{f}(W_i W_i^\top)W_i]_\infty^\#$ is equivalent to

$$\begin{bmatrix} U_{i+1} \\ V_{i+1} \end{bmatrix} = \begin{bmatrix} U_i \\ V_i \end{bmatrix} - \frac{\eta_i}{2} \begin{bmatrix} \nabla f(U_i V_i^\top)V_i \\ \nabla f(U_i V_i^\top)^\top U_i \end{bmatrix}_\infty^\# . \quad \text{(V.4)}$$

Moreover, we can relate the smoothness of $\hat{f}$ to that of f; the following lemma generalizes Proposition 3.1. of [18].

Lemma 2. *Let* $\hat{f}\left(\begin{bmatrix} A & B \\ B^\top & D \end{bmatrix}\right) \triangleq f(B)$ *be defined on the PSD cone. Suppose that* f *is convex and* L*-smooth in some Schatten-*p *norm:*

$$\|\nabla f(X) - \nabla f(Y)\|_{S_q} \leq L\|Y - X\|_{S_p}, \tag{V.5}$$

where q, p *satisfies* $\frac{1}{p}+\frac{1}{q} = 1$*. Suppose also that* $\nabla f(\cdot) \in \mathbb{R}^{p\times p}$ *is symmetric. Then* $\hat{f}$ *is convex and* L*-smooth in the same norm; i.e.,* $\hat{f}$ *also satisfies* (V.5).

Proof. See Section F in the supplementary material. □

Hence, if we apply (V.3) (resp. (V.4)) to the lifted objective (V.2), then the results in Section IV-A (resp. Section III-A) hold (with obvious changes in the constants).

VI. Experiments

Two real-world applications are considered: Fourier Ptychography and text classification via the `FastText` architecture. The former application has a PSD-constrained objective, and the latter unconstrained. We show that the tensor-based Algorithm 3 exploits the low-rank structure of natural images, and hence leads to state-of-the-art performance on synthetic and real data. For the latter application, we show that spectral gradient descent leads to considerable speedups.

A. Fourier Ptychography

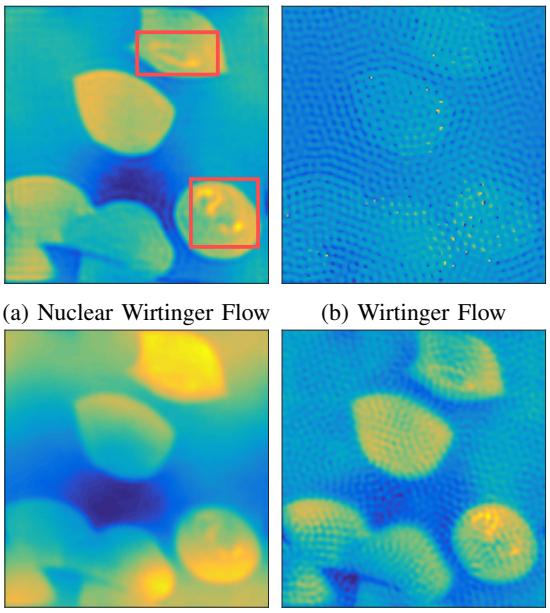
We consider the task of *Fourier Ptychography* reconstruction, a computational imaging technique that aims to reconstruct a high-resolution image based on a collection of low-resolution samples [53], [54]. Ptychography reconstruction is subclass of Phase Retrieval, and the factorized gradient method for such applications has the domain name *Wirtinger Flow* [6], with rank parameter $r = 1$.

We consider Algorithm 3, henceforth referred to as *nuclear Wirtinger flow*, for ptychography reconstruction. As a baseline comparison, we first perform extensive comparison against existing algorithms for synthetic data in Section VI-A1. In Section VI-A2, we show that the nuclear Wirtinger flow is the only algorithm that succeeds for detecting malaria infection in a reasonable amount of time.

1) Synthetic Data: We adopt the same setting as the online library *PhasePack* [55]: In (III.1), we choose a_i's from empirical measurements obtained by an optical device [56]. A synthetic image is passed through these measurements using (III.1), and we report the images of various algorithms returned in 5 minutes. We perform parameter sweeping for all recovery algorithms to find the best setting.

The results are reported in Figure 1; for more results, see Section G in the supplementary material.

Since the true image in the synthetic data is simple, many of the non-convex algorithms, including Wirtinger flow and nuclear Wirtinger flow, succeed in recovering the image quickly and yield comparable results. On the other hand, the convex methods, such as the SketchyCGM [50], only return noisy figures given limited time. The Truncated Amplitude Flow method, while known to perform well in the coded diffraction model [49], fails to recover even simple images in the ptychographic reconstruction.



(a) Nuclear Wirtinger Flow (b) Wirtinger Flow

(c) SketchyCGM (d) Fienup method

Fig. 2: Fourier ptychography reconstructions.

2) Real Data: We use the real dataset provided by the authors of [54]. The dataset consists of Fourier ptychographic measurements taken from patients with malaria infection, where the number of samples is 185600 and the image to be recovered contains 6400 pixels. The critical task is to obtain reconstructions that allow clear identification of the infected cells. The objective function for Fourier ptychography falls under the category of (II.11); we adopt the same setups as in [54], and we refer the readers to the reference for details.

We incorporated four algorithms: the Wirtinger and nuclear Wirtinger flow, which are the best-performing non-convex methods in Section VI-A1, the Fienup [44], a classical method in computational imaging, and SketchyCGM [50], a convex algorithm that directly solves the unfactorized problem (III.3). The step-sizes are obtained by parameter sweeping.

Figure 2 presents the reconstruction images from various methods. All implementations are in MATLAB. We take 20 random initializations for the Wirtinger and nuclear Wirtinger flow, and we report the best reconstruction. We run 1000 iterations for all the algorithms except for Fienup (also known as "Alternating Projections" in [54]), for which the reconstruction is provided by the authors of [54] without

Table 1: Time comparison

Algorithm	Time (sec.)
Wirtinger Flow	13.7799
SketchyCGM	1.6038e+03s
Nuclear Wirtinger Flow	22.8751

implementation details.

From Figure 2, we see that the infected cells are clearly visible in the nuclear Wirtinger Flow as boxed in red. The SketchyCGM algorithm, as a convex method, is time-consuming but fairly robust, and it returns the second best image in terms of quality. However, the infected cells are barely visible from the reconstruction, even though it takes 70 times as the nuclear Wirtinger flow (cf., Table 1). The Fienup and Wirtinger Flow produce serious artifact, and the latter completely fails to recover the image.

Table 1 compares the running time for all the algorithms except for Fienup. We observe that our method results in 66% computational time overhead compared to Wirtinger flow, but the overall running time is still reasonably short. On the other hand, while we expect SketchyCGM to recover the same quality of the image as the nuclear Wirtinger flow, provided that we run it for more iterations, the high-dimensional nature of the problem renders the running time fairly slow.

A more detailed comparison between nuclear Wirtinger Flow and Wirtinger Flow can be found in Section H of the supplementary material.

B. Text classification by `FastText`

Text classification is one of the most important tasks in Natural Language Processing. Recently, a simple model, called `FastText`, has been proposed to solve text classification problems for very large corpus with large output space. The `FastText` assumes that the input-output relation of text classification can be explained by a large matrix $C \in \mathbb{R}^{p\times q}$, and the objective is to minimize the log-softmax output over training data $\{(x_n, y_n)\}_{n=1}^N$:

$$\min_{C\in\mathbb{R}^{p\times q}} -\frac{1}{N}\sum_{n=1}^{N} y_n \log f(Cx_n). \tag{VI.1}$$

The key idea of `FastText` is to fix a small intermediate value r, and decompose $C = AB^\top$ where $A \in \mathbb{R}^{p\times r}$ and $B \in \mathbb{R}^{q\times r}$. The role of r is twofold: First, it speeds up the training process by constraining the decision variable to small rank. Second, it prevents overfitting due to the excessive number of parameters in the large matrix C. We refer to [42], [43] for further details.

The main component of the objective in (VI.1) is the matrix-variate lse in (IV.7). Motivated by the results in Section IV-B, we propose to run Algorithm (V.4) for the factorized program of (VI.1), with r fixed to 10. From (VI.1), one can infer that computing the gradient takes $O\left(pqr + rN\min\{p,q\}\right)$ time, and hence the overhead of the $[\cdot]_\infty^\#$ operation (which takes $O\left(r^2\min\{p,q\}\right)$) is negligible, as $N \gg r$.

We test the iterate (V.4) (the red curve in Fig. 3 and 5) on 6 datasets whose information can be found in [57]. The baseline we compare to is the gradient descent algorithm (the black curve in Fig. 3 and 5) proposed in [43]. We have also included a heuristic approximation of the iterates (V.4) (the blue curve in Fig. 3 and 5), by employing $\begin{bmatrix}A\\B\end{bmatrix}_\infty^\# \simeq \begin{bmatrix}[A]_\infty^\#\\[B]_\infty^\#\end{bmatrix}$.

All the experiments are implemented in C++, and run on the processor Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz. Learning rates for each of the algorithms are obtained through 5-fold cross-validation.

Figure 3 shows the training and test performance on two datasets. The heuristic version of (V.4) performs the best in terms of training errors. However, the theoretical iterate (V.4) generalizes the best. In all cases, the spectral iterates outperform the classic gradient descent. These observations are consistent throughout our experiments; see Section I of the supplementary material for more evidence.

VII. Conclusion

This paper introduces a non-Euclidean, first-order methods into the factorization framework for solving (I.1). The framework is easy to implement. We provide rigorous convergence rates, under assumptions akin to the classical gradient methods. We demonstrate the empirical success of our algorithms on phase retrieval and text classification.

We would like to note that for the phase retrieval application, there is a growing literature of algorithms, with different speed enhancements. We note that with the additional twists, many of these state-of-the-art methods perform well when applied to synthetic data. However, we observe that they have major robustness issues in real data, possibly due to imperfect calibration of the linear measurements. We believe the simplicity of our algorithm is a strength in this setting even though it can be enhanced with additional tricks, such as reshaping, truncation, hybrid, and minibatch [58]–[60], which is beyond the scope of this initial study.

As a result, we have established a different, but very strong baseline for our comparisons: The first scalable convex optimization approach [50], which none of the non-convex methods include. We show that convex method indeed outperforms the other non-convex approaches in the literature in terms of the solution quality (but certainly NOT speed!). However, our new algorithm still outperforms the convex method, while being similar in speed to other fast non-convex methods, such as the Wirtinger flow [6].

In the `FastText` application, the spectral norm provides state-of-the-art results with nearly orthogonal factors in the respective space. We leave the interpretation of this result as future work.

VIII. Acknowledgment

The authors would like to gratefully acknowledge Roarke Horstmeyer for providing the Fourier Ptychography data.

This work was supported in part by the European Commission (ERC 725594 [time-data]), the Hasler Foundation (project no. 16066), and the Swiss National Science Foundation (SNSF) under grant number 407540_167319.

References

[1] S. Aaronson, "The learnability of quantum states," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2088. The Royal Society, 2007, pp. 3089–3114.

[2] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing," *Physical review letters*, vol. 105, no. 15, p. 150401, 2010.

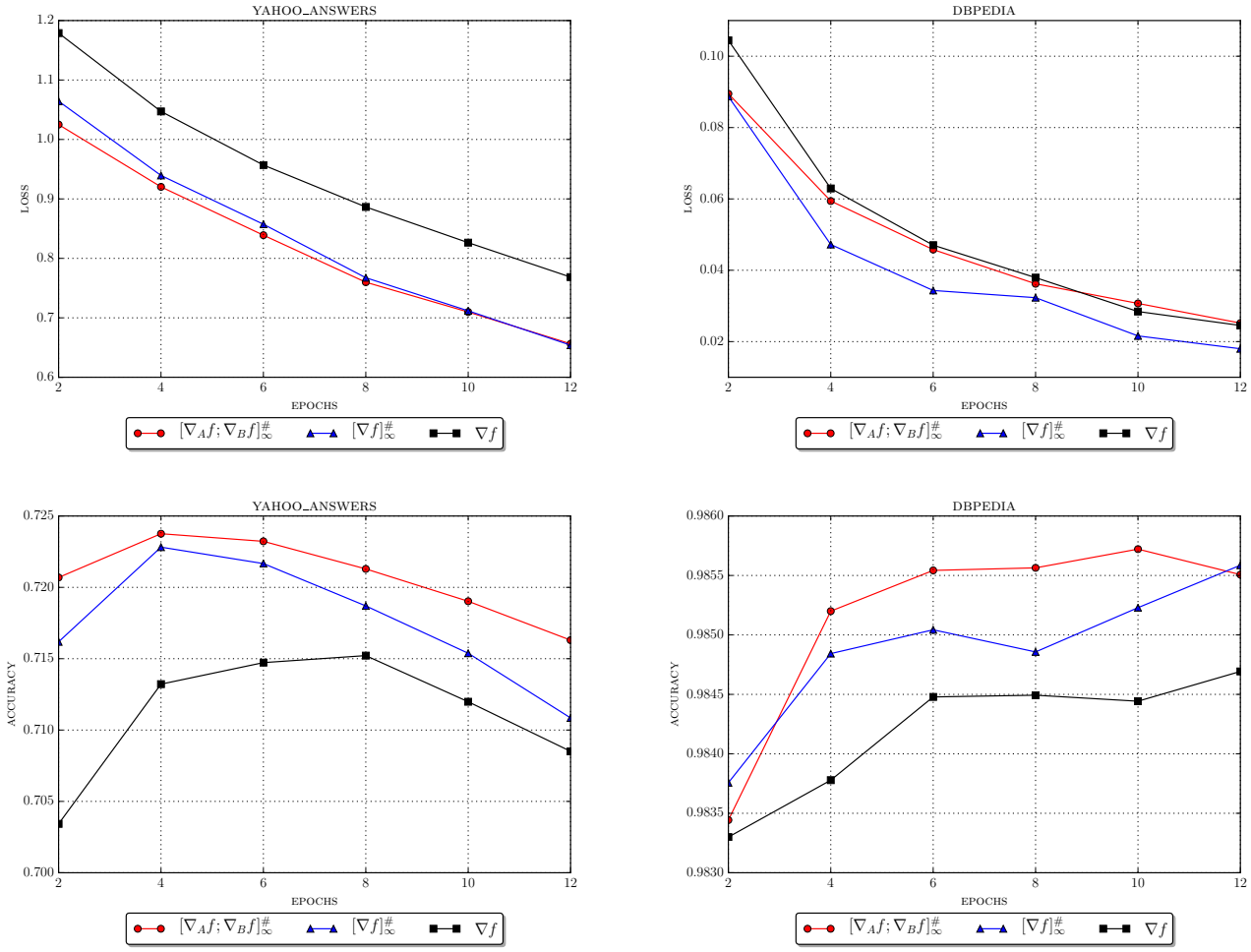


Fig. 3: Top row: Training loss for `YahooAnswers`, and test accuracy for `YahooAnswers`. Bottom row: Training loss for `dbpedia`, and test accuracy for `dbpedia`.

[3] C.-J. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik, "Sparse inverse covariance matrix estimation using quadratic approximation," in *Advances in neural information processing systems*, 2011, pp. 2330–2338.

[4] A. Kyrillidis, R. K. Mahabadi, Q. T. Dinh, and V. Cevher, "Scalable sparse covariance estimation via self-concordance," in *28th AAAI Conference on Artificial Intelligence, AAAI 2014, 26th Innovative Applications of Artificial Intelligence Conference, IAAI 2014 and the 5th Symposium on Educational Advances in Artificial Intelligence, EAAI 2014*. AI Access Foundation, 2014.

[5] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 2796–2804.

[6] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[7] L. Bian, J. Suo, G. Zheng, K. Guo, F. Chen, and Q. Dai, "Fourier ptychographic reconstruction using wirtinger flow optimization," *Optics express*, vol. 23, no. 4, pp. 4856–4866, 2015.

[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[9] N. Srebro, J. D. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization." in *NIPS*, vol. 17, 2004, pp. 1329–1336.

[10] K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. S. Dhillon, and A. Tewari, "Prediction and clustering in signed networks: a local to global perspective." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1177–1213, 2014.

[11] 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*, vol. 12, no. 2, p. e1004760, 2016.

[12] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[13] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 2013, pp. 665–674.

[14] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 4. IEEE, 2004, pp. 3273–3278.

[15] 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 transactions on automation science and engineering*, vol. 3, no. 4, pp. 360–371, 2006.

[16] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," *Mathematical Programming*, vol. 149, no. 1-2, pp. 47–81, 2015.

[17] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi, "Dropping convexity for faster semi-definite optimization," in *29th Annual Conference on Learning Theory*, 2016, pp. 530–582.

[18] D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi, "Finding low-rank solutions to matrix problems, efficiently and provably," *arXiv preprint arXiv:1606.03168*, 2016.

[19] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 427–435.

[20] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford, "An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2014, pp. 217–226.

[21] D. E. Carlson, V. Cevher, and L. Carin, "Stochastic spectral descent for

restricted boltzmann machines." in *AISTATS*, 2015.
[22] D. E. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher, "Preconditioned spectral descent for deep learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 2953–2961.
[23] D. Carlson, Y.-P. Hsieh, E. Collins, L. Carin, and V. Cevher, "Stochastic spectral descent for discrete graphical models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 296–311, 2016.
[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.
[25] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.
[26] ——, "Local minima and convergence in low-rank semidefinite programming," *Mathematical Programming*, vol. 103, no. 3, pp. 427–444, 2005.
[27] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via non-convex factorization," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6535–6579, 2016.
[28] 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.
[29] 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 Machine Learning*, 2016, pp. 964–973.
[30] M. Hardt and M. Wootters, "Fast matrix completion without the condition number." in *COLT*, 2014, pp. 638–678.
[31] Q. Zheng and J. Lafferty, "Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent," *arXiv preprint arXiv:1605.07051*, 2016.
[32] D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi, "Non-square matrix sensing without spurious local minima via the burer-monteiro approach," *arXiv preprint arXiv:1609.03240*, 2016.
[33] Y. Chen and M. J. Wainwright, "Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees," *arXiv preprint arXiv:1509.03025*, 2015.
[34] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
[35] S. Sra and R. Hosseini, "Conic geometric optimization on the manifold of positive definite matrices," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 713–739, 2015.
[36] N. Boumal, P.-A. Absil, and C. Cartis, "Global rates of convergence for nonconvex optimization on manifolds," *arXiv preprint arXiv:1605.08101*, 2016.
[37] 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.
[38] I. Cioranescu, *Geometry of Banach spaces, duality mappings and nonlinear problems*. Springer Science & Business Media, 2012, vol. 62.
[39] H. Jarchow, *Locally convex spaces*. Springer Science & Business Media, 2012.
[40] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.
[41] ——, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.
[42] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.
[43] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.
[44] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.
[45] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.
[46] O. Dhifallah, C. Thrampoulidis, and Y. M. Lu, "Phase retrieval via linear programming: Fundamental limits and algorithmic improvements," *arXiv preprint arXiv:1710.05234*, 2017.
[47] S. Bahmani and J. Romberg, "Phase retrieval meets statistical learning theory: A flexible convex relaxation," in *Artificial Intelligence and Statistics*, 2017, pp. 252–260.
[48] T. Goldstein and C. Studer, "Phasemax: Convex phase retrieval via basis pursuit," *arXiv preprint arXiv:1610.07531*, 2016.
[49] G. Wang, G. B. Giannakis, and Y. C. Eldar, "Solving systems of random quadratic equations via truncated amplitude flow," *IEEE Transactions on Information Theory*, 2017.
[50] A. Yurtsever, M. Udell, J. A. Tropp, and V. Cevher, "Sketchy decisions: Convex low-rank matrix optimization with optimal storage," in *20th International Conference on Artificial Intelligence and Statistics (AISTATS2017)*, no. EPFL-CONF-225653, 2017.
[51] Z. Yuan and H. Wang, "Phase retrieval via reweighted wirtinger flow," *Applied optics*, vol. 56, no. 9, pp. 2418–2427, 2017.
[52] M. Jaggi, M. Sulovsk *et al.*, "A simple algorithm for nuclear norm regularized problems," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 471–478.
[53] G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution fourier ptychographic microscopy," *Nature photonics*, vol. 7, no. 9, pp. 739–745, 2013.
[54] R. Horstmeyer, R. Y. Chen, X. Ou, B. Ames, J. A. Tropp, and C. Yang, "Solving ptychography with a convex relaxation," *New journal of physics*, vol. 17, no. 5, p. 053044, 2015.
[55] R. Chandra, C. Studer, and T. Goldstein, "Phasepack: A phase retrieval library," *arXiv preprint arXiv:1711.10175*, 2017.
[56] C. A. Metzler, M. K. Sharma, S. Nagesh, R. G. Baraniuk, O. Cossairt, and A. Veeraraghavan, "Coherent inverse scattering via transmission matrices: Efficient phase retrieval algorithms and a public dataset," in *Computational Photography (ICCP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–16.
[57] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.
[58] H. Zhang, Y. Chi, and Y. Liang, "Provable non-convex phase retrieval with outliers: Median truncatedwirtinger flow," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1022–1031.
[59] R. Kolte and A. Özgür, "Phase retrieval via incremental truncated wirtinger flow," *arXiv preprint arXiv:1606.03196*, 2016.
[60] G. Wang, L. Zhang, G. B. Giannakis, M. Akçakaya, and J. Chen, "Sparse phase retrieval via truncated amplitude flow," *arXiv preprint arXiv:1611.07641*, 2016.