Tech. report UCL-INMA-2016.02-v2

Solving PhaseLift by Low-rank Riemannian Optimization Methods*

Wen Huang[1], Kyle A. Gallivan[2], and Xiangxiong Zhang[3]

[1] ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium.
[2] Department of Mathematics, Florida State University, Tallahassee FL, USA.
[3] Department of Mathematics, Purdue University, West Lafayette, IN, USA.

Abstract

A framework, PhaseLift, was recently proposed to solve the phase retrieval problem. In this framework, the problem is solved by optimizing a cost function over the set of complex Hermitian positive semidefinite matrices. This paper considers an approach based on an alternative cost function defined on a union of appropriate manifolds. It is related to the original cost function in a manner that preserves the ability to find a global minimizer and is significantly more efficient computationally. A rank-based optimality condition for stationary points is given and optimization algorithms based on state-of-the-art Riemannian optimization and dynamically reducing rank are proposed. Empirical evaluations are performed using the PhaseLift problem. The new approach is shown to be an effective method of phase retrieval with computational efficiency increased substantially compared to the algorithm used in original PhaseLift paper.

1 Introduction

Recovering a signal given the modulus of its transform, e.g., Fourier or wavelet transform, is an important task in the phase retrieval problem. It is a key problem for many important applications, e.g., X-ray crystallography imaging [14], diffraction imaging [7], optics [26] and microscopy [20].

This paper considers the discrete form of the phase retrieval problem where an indexed set of complex numbers $\mathbf{x} \in \mathbb{C}^{n_1 \times n_2 \times \ldots \times n_s}$ is to be recovered from the modulus of its discrete Fourier transform $|\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s)|$, where $(g_1, g_2, \ldots, g_s) \in \Omega := G_1 \times G_2 \times \ldots G_s$ and Ω is a grid in an s-dimensional space. The discrete Fourier transform of $\mathbf{x}$, denoted $\tilde{\mathbf{x}}$, is given by

$$\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s) = \\ \frac{1}{\sqrt{n}} \sum_{i_1, i_2, \ldots, i_s} \mathbf{x}_{i_1 i_2 \ldots i_s} \exp\left(-2\pi \left(\frac{(i_1 - 1)g_1}{n_1} + \ldots + \frac{(i_s - 1)g_s}{n_s}\right)\sqrt{-1}\right), \tag{1.1}$$

where $n = n_1 n_2 \ldots n_s$, i_j is an integer satisfying $1 \leq i_j \leq n_j$ for $j = 1, \ldots s$, $\mathbf{x}_{i_1 i_2 \ldots i_s}$ denotes the corresponding entry of $\mathbf{x}$ and $\tilde{\mathbf{x}}(g_1, g_2, \ldots, g_s)$ denotes the corresponding entry of $\tilde{\mathbf{x}}$.

It is well-known that the solution of the phase retrieval problem is not unique. Many approaches e.g., [23, 13, 9, 25, 11] have been proposed to recover the phase. Some frameworks use multiple structured illuminations or the mathematically equivalent construct of masks combined with convex programming, e.g., PhaseLift [9]. For the PhaseLift framework, four major results are of interest here. First, using a small number (related to s) of noiseless measurements of the

*This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by grant FNRS PDR T.0173.13.

modulus defined by certain carefully designed illuminations, the phase can be recovered exactly [9]. Second, when these carefully designed measurements are not used, exact recovery is still possible using $O(n^2)$ noiseless measurements [10]. Third, the phase can be recovered exactly with high probability using $O(n \log n)$ noiseless measurements of the modulus [12]. Finally, the stability of recovering the phase using noisy measurements is shown in [12].

The problems in PhaseLift concern optimizing convex cost functions defined on a convex set of complex matrices, i.e.,

$$\min_{X \in \mathcal{D}_n} H(X), \tag{1.2}$$

where $H : \mathcal{D}_n \to \mathbb{R} : X \mapsto H(X)$, and $\mathcal{D}_n$ denotes the set of all n-by-n complex Hermitian positive semidefinite matrices. However, the dimension of (1.2) is usually too large to be solved by standard convex programming techniques. Since the desired optimum, X_*, is known to be a rank-one matrix, a low-rank matrix approximation of the argument matrix is used in [9] to save computations for PhaseLift. While this approximation has good empirical performance, no convergence proof is given in [9].

This paper focuses on the framework of PhaseLift and an alternate cost function $F : \mathbb{C}^{n \times p} \to \mathbb{R} : Y \mapsto F(Y) = H(YY^*)$ defined by matrix factorization is considered. Even though F is not convex, it is shown to be a suitable replacement of the cost function H. Riemannian optimization methods on an appropriate quotient space are used for optimizing F. Using the cost function F with a small dimension p reduces storage and the computational complexity of each iteration. This new approach is shown to perform empirically much better than the low-rank approximate version of the algorithm used for PhaseLift in [9] from the points of view of efficiency and effectiveness. Finally, note that the analysis and algorithm presented is not specific to the cost function used for phase retrieval in PhaseLift but for a general cost function defined on $\mathcal{D}_n$ and therefore the approach has potential for optimization in other applications where the global optimum is known to have low rank. The idea of using low-rank factorization to solve positive semidefinite constrained problems is, of course, not new but all the research results of which the authors are aware, are for real positive semidefinite matrix constraints, see [8, 19].

The paper is organized as follows. Section 2 presents the notation used. The derivation of the optimization problem framework in PhaseLift is given in Section 3. The alternate cost function and optimality conditions are given in Section 4. Riemannian optimization methods and the required geometric objects are presented in Section 5. In Section 6, the effectiveness of the methods are demonstrated and, finally, conclusions are given in Section 7.

2 Notation

For any $\mathbf{z} \in \mathbb{C}^{n_1 \times n_2 \times \dots n_s}$, $\mathrm{vec}(\mathbf{z}) \in \mathbb{C}^n$, where $n = n_1 n_2 \dots n_s$, denotes the vector form of $\mathbf{z}$, i.e., $(\mathrm{vec}(\mathbf{z}))_k = \mathbf{z}_{i_1 i_2 \dots i_s}$, where $k = i_1 + \sum_{j=1}^{s-1} n_1 n_2 \dots n_j (i_{j+1} - 1)$. Given $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{C}^{n_1 \times n_2 \times \dots n_s}$, $\langle \mathbf{z}_1, \mathbf{z}_2 \rangle$ denotes $\mathrm{vec}(\mathbf{z}_1)^T \mathrm{vec}(\mathbf{z}_2)$. $\mathrm{Re}(\cdot)$ denotes the real part of the argument and superscript $*$ denotes the conjugate transpose operator. Given a vector v with length h, all upper case $\mathrm{DIAG}(v)$ denotes an h-by-h diagonal matrix the diagonal entries of which are v. All lower case $\mathrm{diag}(M)$ denotes a vector of the diagonal entries of $M \in \mathbb{C}^{s \times k}$ and $\mathrm{trace}(M)$ denotes the trace of M. If $s \geq k$, $M_\perp$ denotes an $s \times (s-k)$ matrix such that $M_\perp^* M_\perp = I_{(s-k)\times(s-k)}$ and $M_\perp^* M = 0_{(s-k)\times k}$.

Given an embedded submanifold $\mathcal{M} \subseteq \mathbb{C}^{s \times k}$, $\mathrm{T}_x \mathcal{M}$ and $\mathrm{N}_x \mathcal{M}$ denote the tangent s-pace and normal space of $\mathcal{M}$ at $x \in \mathcal{M}$ respectively. $\mathcal{D}_k$ denotes set $\{X \in \mathbb{C}^{n \times n} | X = X^*, X \geq 0, \mathrm{rank}(X) \leq k\}, 1 \leq k \leq n$. Note that the statement $X \geq 0$ means that matrix

X is positive semidefinite or definite. $\mathrm{St}(k,s)$ denotes the complex compact Stiefel manifold $\{A \in \mathbb{C}^{s\times k} | A^*A = I_{k\times k}\}$ with $s \geq k$. $\mathrm{S}_+^{\mathbb{C}}(k,s)$ denotes the set of all Hermitian positive semidefinite $s \times s$ matrices of fixed rank k. $\mathbb{C}_*^{s\times k}$ denotes the complex noncompact Stiefel manifold, i.e., the set of all $s \times k$ full column rank complex matrices. $\mathcal{O}_s$ denotes the group of s-by-s unitary matrices.

3 The PhaseLift Approach to Phase Retrieval

The phase retrieval problem recovers $\mathbf{x}$ from quadratic measurements of the form $\mathbb{A}(\mathbf{x}) = \{|\langle \mathbf{a}_k, \mathbf{x}\rangle|^2 : k = 1, 2, \ldots, m\}$, where $\mathbf{a}_k \in \mathbb{C}^{n_1\times n_2 \times \ldots n_s}, k = 1, 2, \ldots, m$ are given. It is well-known that the quadratic measurements can be lifted up to be linear measurements of the rank-one matrix $X = xx^*$, where $x = \mathrm{vec}(\mathbf{x}) \in \mathbb{C}^n$. Specifically, the measurements are $|\langle \mathbf{a}_k, \mathbf{x}\rangle|^2 = \mathrm{trace}(a_k a_k^* x x^*) := \mathrm{trace}(A_k X)$, where $a_k = \mathrm{vec}(\mathbf{a}_k) \in \mathbb{C}^n$. Define $\mathcal{A}$ to be the linear operator mapping X into $b := \{\mathrm{trace}(A_k X) : k = 1, 2, \ldots m\}$. The goal of the phase retrieval problem is to

$$\text{find } X, \quad \text{such that } \mathcal{A}(X) = b, X \geq 0 \text{ and } \mathrm{rank}(X) = 1. \tag{3.1}$$

The alternative problem suggested in [9] considers an optimization problem that does not force the rank of matrix to be one but adds a nuclear norm penalty term to favor low-rank solutions

$$\min_{X\in\mathcal{D}_n} \|b - \mathcal{A}(X)\|_2^2 + \kappa\, \mathrm{trace}(X), \tag{3.2}$$

where κ is a positive constant.

Measurements with noise, $b \in \mathbb{R}^m$, are assumed to have the form $b = \mathcal{A}(X) + \epsilon$, where $\epsilon \in \mathbb{R}^m$ is noise sampled from a distribution $p(:;\mu)$. The task suggested in [9] is

$$\begin{aligned} &\min_X - \log(p(b;\mu)) + \kappa\, \mathrm{trace}(X) \\ &\text{such that } \mu = \mathrm{diag}(ZXZ^*) \text{ and } X \in \mathcal{D}_n, \end{aligned} \tag{3.3}$$

or equivalently

$$\min_{X\in\mathcal{D}_n} - \log(p(b; \mathrm{diag}(ZXZ^*))) + \kappa\, \mathrm{trace}(X) \tag{3.4}$$

where κ is a positive constant. Problems (3.3) and (3.4) are preferred over Problem (3.1), since they are convex programming problems when the log-likelihood function is concave.

4 Theoretical Results

This section presents theoretical results that motivate the design of algorithms for optimizing a class of cost functions defined on $\mathcal{D}_n$ without giving the proofs due to space limits. They can be found in [18]. The analysis does not rely on the convexity of the particular cost function H from the class.

4.1 Equivalent Cost Function

The cost functions generically denoted H all satisfy

$$H : \mathcal{D}_n \to \mathbb{R} : X \mapsto H(X). \tag{4.1}$$

It is well-known that for any $X \in \mathcal{D}_n$, there exists $Y_n \in \mathbb{C}^{n\times n}$ such that $Y_n Y_n^* = X$. Furthermore, if X has rank p, then there exists $Y_p \in \mathbb{C}^{n\times p}$ such that $Y_p Y_p^* = X$. Throughout this paper, the subscript of Y is used to denote the column size of Y. A surjective mapping between $\mathbb{C}^{n\times p}$ and $\mathcal{D}_p$ is given by $\alpha_p : \mathbb{C}^{n\times p} \to \mathcal{D}_p : Y_p \mapsto Y_p Y_p^*$. Thus, if the desired solution of H is known to be at most rank p, then an alternate cost function to H can be used:

$$F_p : \mathbb{C}^{n\times p} \to \mathbb{R} : Y_p \mapsto H(\alpha_p(Y_p)) = H(Y_p Y_p^*).$$

The subscripts of F and α indicate the column size of the argument. The domain of F_p has lower dimension than that of H which may yield computational efficiency. Therefore, instead of Problem (1.2), the problem $\min_{Y_p \in \mathbb{C}^{n\times p}} F_p(Y_p)$ is considered.

4.2 Optimality Conditions

In this section, characterizations of stationary points of F and H over $\mathcal{D}_n$ are used to derive the relationship between optimizing F and optimizing H over $\mathcal{D}_n$. Since H is defined on a constrained set, a stationary point of H does not simply satisfy $\operatorname{grad} H(X) = 0$. The stationary points of H are defined as follows by [18, Lemma 5]:

Definition 4.1. *A stationary point of* (4.1) *is a matrix* $X \in \mathbb{D}_n$ *such that* $\operatorname{grad} H(X)X = 0$ *and* $\operatorname{grad} H(X) \geq 0$.

The gradient and the action of Hessian of F_p are easily computed and are given in Lemma 4.1 in terms of H.

Lemma 4.1. *The gradient of* F_p *at* Y_p *is given by* $\operatorname{grad} F_p(Y_p) = 2 \operatorname{grad} H(Y_p Y_p^*) Y_p$ *and the action of the Hessian of* F_p *at* Y_p *on* $\eta_p \in \mathbb{C}^{n\times p}$ *is given by* $\operatorname{Hess} F_p(Y_p)[\eta_p] = 2 \operatorname{grad} H(Y_p Y_p^*)\eta_p + 2(\operatorname{Hess} H(Y_p Y_p^*)[\eta_p Y_p^* + Y_p \eta_p^*])Y_p$.

Theorem 4.1 and [19, Theorem 7] show similar results under different frameworks. Both results suggest considering the cost function F_p if the desired minimizer of H is known to have rank smaller than p, as is the case with PhaseLift for phase retrieval. This is formalized in Theorem 4.1 and has critical algorithmic, efficiency and optimality implications when H has suitable structure such as convexity as in the case of PhaseLift.

Theorem 4.1. *Suppose* $Y_p = K_s Q^*$ *is a rank deficient minimizer of* F_p*, where* $K_s \in \mathcal{C}_*^{n\times s}$ *and* $Q \in \operatorname{St}(s,p)$*. Then* $(K_s)_\perp^* \operatorname{grad} H(Y_p Y_p^*)(K_s)_\perp$ *is a positive semidefinite matrix and, therefore,* $X = Y_p Y_p^*$ *is a stationary point of* H*. If furthermore* H *is convex, then* X *is a global minimizer of* (4.1).

5 A Riemannian Approach

Riemannian optimization is an active research area and recently many Riemannian optimization methods have been systemically analyzed and efficient libraries designed, e.g., Riemannian trust-region Newton method [1, 4], Riemannian Broyden family of methods including RBFGS and its limited-memory version (LRBFGS) [22, 15, 17], Riemannian trust-region symmetric rank-one update method and its limited-memory version [15, 16], Riemannian Newton method and Riemannian non-linear conjugate gradient method [24].

5.1 Riemannian Optimization on Fixed Rank Manifold

Derivations for Riemannian objects of $\mathrm{S}_{+}^{\mathbb{R}}(p,n)$ have been given in [2]. This section includes results of Riemannian objects for the complex case, i.e., $\mathrm{S}_{+}^{\mathbb{C}}(p,n)$. Since the mapping α_p is not an injection, all the minimizers of F_p are degenerate, which causes difficulties in some algorithms, e.g., Riemannian and Euclidean Newton method. In order to overcome this difficulty, a function defined on a quotient manifold with fixed rank is considered. To this end, define the mapping β_p to be the mapping α_p restricted to $\mathbb{C}_*^{n\times p}$, i.e., $\beta_p : \mathbb{C}_*^{n\times p} \to \mathrm{S}_{+}^{\mathbb{C}}(p,n) : Y \mapsto \alpha_p(Y) = YY^*$. and the function G_p to be F_p restricted to $\mathbb{C}_*^{n\times p}$, i.e., $G_p : \mathbb{C}_*^{n\times p} \to \mathbb{R} : Y \mapsto F_p(Y) = H(\beta_p(Y))$. Like α_p, the mapping β_p is a surjection but not a injection and there are multiple matrices in $\mathbb{C}_*^{n\times p}$ mapping to a single point in $\mathrm{S}_{+}^{\mathbb{C}}(p,n)$. Nevertheless, given a $X \in \mathrm{S}_{+}^{\mathbb{C}}(p,n)$, $\beta_p^{-1}(X)$ is a manifold while $\alpha_p^{-1}(X)$ is not a manifold. Therefore, using the mapping β_p, a quotient manifold can be used to remove the degeneracy by defining the equivalence class $\beta_p^{-1}(YY^*) = [Y] = \{YO | O \in \mathcal{O}_p\}$ and the set

$$\mathbb{C}_*^{n\times p}/\mathcal{O}_p = \{[Y] | Y \in \mathbb{C}_*^{n\times p}\}.$$

This set can be shown to be a quotient manifold over $\mathbb{R}$. To clarify the notation, $\pi(Y)$ is used to denote $[Y]$ viewed as an element in $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $\pi^{-1}(\pi(Y))$ is used to denote $[Y]$ viewed as a subset of $\mathbb{C}_*^{n\times p}$. The function $m_p : \pi(Y) \mapsto YY^*$ is a diffeomorphism between $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ and $\mathrm{S}_{+}^{\mathbb{C}}(p,n)$.

Choosing a representative for an equivalence class and definitions of related mathematical objects have been developed in many papers in the literature of computation on manifolds, e.g., [3]. The vertical space at $Y \in \pi^{-1}(\pi(Y))$, which is the tangent space of $\pi^{-1}(\pi(Y))$ at Y, is

$$\mathcal{V}_Y = \{Y\Omega | \Omega^* = -\Omega, \Omega \in \mathbb{C}^{p\times p}\}.$$

The horizontal space at Y, $\mathcal{H}_Y$, is defined to be a subspace of $\mathrm{T}_Y\,\mathbb{C}_*^{n\times p} = \mathbb{C}^{n\times p}$ that is orthogonal to $\mathcal{V}_Y$, i.e., satisfying $\mathcal{H}_A \oplus \mathcal{V}_A = \mathrm{T}_A\,\mathrm{GL}(n,\mathbb{C})$. Therefore, a Riemannian metric of $\mathbb{C}_*^{n\times p}$ is required to define the meaning of orthogonal. The standard Euclidean metric,

$$g_Y(\eta_Y, \xi_Y) = \mathrm{Re}(\mathrm{trace}(\eta_Y^* \xi_Y)) \tag{5.1}$$

for all $\eta_Y, \xi_Y \in \mathrm{T}_Y\,\mathbb{C}_*^{n\times p}$ and $Y \in \mathbb{C}_*^{n\times p}$, is used in this paper. The horizontal space is therefore

$$\begin{aligned}\mathcal{H}_Y =& \{V \in \mathbb{C}^{n\times p} | Y^*V = V^*Y\}\\ =& \{Y(Y^*Y)^{-1}S + Y_\perp K | S^* = S, S \in \mathbb{C}^{p\times p}, K \in \mathbb{C}^{(n-p)\times p}\}.\end{aligned}$$

The horizontal space $\mathcal{H}_Y$ is a representation of the tangent space $\mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$.

It is known that for any $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)}\,\mathbb{C}_*^{n\times p}/\mathcal{O}_p$, there exists a unique vector in $\mathcal{H}_Y$, called the horizontal lift of $\eta_{\pi(Y)}$ and denoted by $\eta_{\uparrow_Y}$, satisfying $\mathrm{D}\,\pi(Y)[\eta_{\uparrow_Y}] = \eta_{\pi(Y)}$, see e.g., [3]. The orthogonal projections onto the horizontal space or the vertical space are also easily characterized.

Lemma 5.1. *The orthogonal projection onto the vertical space $\mathcal{V}_Y$ of $\eta \in \mathbb{C}_*^{n\times p}$ is $P_Y^v(\eta) = Y\Omega$, where Ω is the skew symmetric matrix that solves the Sylvester equation, $\Omega Y^*Y + Y^*Y\Omega = Y^*\eta - \eta^*Y$. The orthogonal projection onto the horizontal space $\mathcal{H}_Y$ is $P_Y^h(\eta) = \eta - Y\Omega$.*

Finally, the desired cost function that removes the equivalence is defined as

$$f_p : \mathbb{C}_*^{n\times p}/\mathcal{O}_p \to \mathbb{R} : \pi(Y) \mapsto f_p(\pi(Y)) = G_p(Y) = F_p(Y). \tag{5.2}$$

The function f_p in (5.2) has the important property that $\pi(Y)$ is a nondegenerate minimizer of f over $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$ if and only if YY^* is a nondegenerate minimizer of H over $\mathrm{S}_+^{\mathbb{C}}(p,n)$.

The gradient and the action of the Hessian of (5.2) are given in Lemma 5.2.

Lemma 5.2. *The gradient of f satisfies $(\operatorname{grad} f(\pi(Y)))_{\uparrow_Y} = P_Y^h(\operatorname{grad} F(Y))$, and the action of Hessian of (5.2) at $\pi(Y)$ along $\eta_{\pi(Y)} \in \mathrm{T}_{\pi(Y)}\, \mathbb{C}_*^{n\times p}/\mathcal{O}_p$ satisfies $(\operatorname{Hess} f(\pi(Y))[\eta_{\pi(Y)}])_{\uparrow_Y} = P_Y^h(\dot{M} - \eta_{\uparrow_Y}\Omega)$, where $M = \operatorname{grad} F(Y)$, $\dot{M} = \operatorname{Hess} F(Y)[\eta_{\uparrow_Y}]$, and Ω is the skew-symmetric matrix that solves $\Omega Y^* Y + Y^* Y \Omega = Y^* M - M^* Y$.*

5.2 Dynamic Rank Reduction

The domain of f_p, $\mathbb{C}_*^{n\times p}/\mathcal{O}_p$, is not closed, i.e., a sequence $\{W^{(i)}\}$ representing $\{\pi(W^{(i)})\}$ generated by an algorithm may have a limit point $\hat{W}$ with rank less than p. It is impossible, in practice to check whether a limit point of iterates $\{W^{(i)}\}$ is a lower rank matrix or just close to one of lower rank. However, when the desired rank of the minimizer is known and the current iterate $W^{(i)}$ has a higher rank than the desired rank, as is the case with PhaseLift for phase retrieval, a straightforward technique can be used to address the lack of closure by dynamically reducing the rank. This technique is discussed below.

The thin singular value decomposition of the i-th iterate is $W^{(i)} = U\Sigma V^*$ and $\Sigma = \mathrm{DIAG}(\sigma_1, \sigma_2, \dots, \sigma_p)$, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. Let $\tilde{\sigma}$ be $\|\mathrm{DIAG}(\sigma_1,\dots,\sigma_p)\|_F/\sqrt{p}$. If there exists $q < p$ such that $\sigma_q/\tilde{\sigma} > \delta$ and $\sigma_{q+1}/\tilde{\sigma} \leq \delta$ for a given threshold δ, then $\hat{W} = U(:,1:q)\,\mathrm{DIAG}(\sigma_1,\dots\sigma_q)V(:,1:q)^*$ is chosen to be the initial point for optimizing cost function f_q over $\mathbb{C}_*^{n\times q}/\mathcal{O}_q$. The details of reducing rank are given in Algorithm 1. Note that the step of decreasing the rank may produce an iterate that increases the cost function value. This facilitates global optimization by allowing nondescent steps.

Algorithm 1 Reduce Rank

Input: $Y \in \mathbb{C}^{n\times p}$; threshold δ;
Output: $W \in \mathbb{C}^{n\times q}$;
1: Take thin singular value decomposition for Y, i.e., $Y = U\,\mathrm{DIAG}(\sigma_1,\dots,\sigma_p)V^*$, where $U \in \mathbb{C}^{n\times p}$, $V \in \mathbb{C}^{p\times p}$ and $\sigma_1 \geq \dots \geq \sigma_p \geq 0$;
2: Set $\tilde{\sigma} = \|\mathrm{DIAG}(\sigma_1,\dots,\sigma_p)\|_F/\sqrt{p}$;
3: **if** $\sigma_p/\tilde{\sigma} > \delta$ **then**
4: $\quad q \leftarrow p$, $W \leftarrow Y$ and return;
5: **else**
6: $\quad$ Find q such that $\sigma_q/\tilde{\sigma} > \delta$ and $\sigma_{q+1}/\tilde{\sigma} \leq \delta$;
7: $\quad$ Let $W = U(:,1:q)\,\mathrm{DIAG}(\sigma_1,\dots\sigma_q)V(:,1:q)^*$ and return;
8: **end if**

Combining a Riemannian optimization method with the procedure of reducing rank gives Algorithm 2.

6 Experiments

In this section, numerical simulations for noiseless problems and those with Gaussian noise are used to illustrate the performance of the proposed method and to compare it to the performance of the current convex optimization approach.

Algorithm 2 Rank Reduce Algorithm

Input: $p > 0$; $Y_p^{(0)} \in \mathbb{C}^{n \times p}$ a representation of initial point $\pi(Y_p^{(0)})$ for f; Stopping criterion threshold ϵ; rank reducing threshold δ; a Riemannian optimization method;

Output: W

for $k = 0, 1, 2, \dots$ **do**
  Apply Riemannian method for cost function f over $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$ with initial point $\pi(Y_p^{(k)})$ until i-th iterate $W^{(i)}$ satisfying $g(\operatorname{grad} f, \operatorname{grad} f) < \epsilon^2$ or the requirement of reducing rank with threshold δ;
  if $g(\operatorname{grad} f, \operatorname{grad} f) < \epsilon_1^2$ **then**
    Find a minimizer $W = W^{(i)}$ over $\mathbb{C}_*^{n \times p}/\mathcal{O}_p$ and return;
  else {iterate in the Riemannian optimization method meets the requirements of reducing rank}
    Apply Algorithm 1 with threshold δ and obtain an output $\hat{W} \in \mathbb{C}^{n \times q}$;
    $p \leftarrow q$ and set $Y_p^{(k+1)} = \hat{W}$;
  end if
end for

6.1 Cost Function, Gradient, and Action of Hessian and Complexity for PhaseLift

The known random masks or illumination fields defined on the discrete signal domain are denoted $\mathbf{w}_r \in \mathbb{C}^{n_1 \times n_2 \times \dots n_s}$, $r = 1, \dots l$. It follows that $\{\langle \mathbf{a}_k, \mathbf{x} \rangle, k = 1, \dots m\}$ is

$$\begin{pmatrix} (\mathcal{F}_{n_s} \otimes \mathcal{F}_{n_{s-1}} \otimes \dots \mathcal{F}_{n_1}) \operatorname{DIAG}(w_1) x \\ \vdots \\ (\mathcal{F}_{n_s} \otimes \mathcal{F}_{n_{s-1}} \otimes \dots \mathcal{F}_{n_1}) \operatorname{DIAG}(w_l) x \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product and $\mathcal{F}_{n_i} \in \mathbb{C}^{t_i \times n_i}, i = 1, \dots, s$ denotes the one-dimensional Discrete Fourier Transform (DFT). Let Z_i denote $(\mathcal{F}_{n_s} \otimes \mathcal{F}_{n_{s-1}} \otimes \dots \mathcal{F}_{n_1}) \operatorname{DIAG}(w_i)$, Z denote $(Z_1^T \; Z_2^T \; \dots \; Z_l^T)^T$. We have $\mathbb{A}(\mathbf{x}) = \operatorname{diag}(Z x x^* Z^*)$, which implies that $\mathcal{A}(X) = \operatorname{diag}(Z X Z^*)$.

When the entries in the noise ϵ are drawn from the normal distribution with mean 0 and variance τ, the cost functions of (3.4) and (3.2) are essentially identical, i.e., for (3.2), $H_1(X) = \|b - \operatorname{diag}(ZXZ^*)\|_2^2 + \kappa \operatorname{trace}(X)$, and for (3.4), $H_2(X) = \frac{1}{\tau^2}\|b - \operatorname{diag}(ZXZ^*)\|_2^2 + \kappa \operatorname{trace}(X)$, (see details in [18, Section 3]). Without loss of generality, only the cost function $H(X) = \|b - \operatorname{diag}(ZXZ^*))\|_2^2/\|b\|_2^2 + \kappa \operatorname{trace}(X)$ is considered. The Euclidean gradient of H is $\operatorname{grad} H(X) = \frac{2}{\|b\|_2^2} Z^* \operatorname{DIAG}(\operatorname{diag}(ZXZ^*) - b) Z + \kappa I_{n \times n}$, and the action of the Euclidean Hessian at X along V is $\operatorname{Hess} H(X)[V] = \frac{2}{\|b\|_2^2} Z^* \operatorname{DIAG}(\operatorname{diag}(ZVZ^*)) Z$, where $V = V^*$. The gradients and actions of Hessians of functions F_p and f_p can be constructed using Lemmas 4.1 and 5.2.

The complexities of evaluations of the function value, gradient and action of Hessian of F_p are all of the same order, $O(pms \max_i(\log(n_i))$. The complexities of evaluations of the function value, gradient and action of Hessian of f_p are $O(pms \max_i(\log(n_i))$, $O(pms \max_i(\log(n_i)) + O(np^2) + O(p^3)$ and $O(pms \max_i(\log(n_i)) + O(np^2) + O(p^3)$ respectively. If $p << n$ then all of these complexities are dominated by $O(pms \max_i(\log(n_i))$.

noiseless	Algorithm 2	LR-FISTA				
		1	2	4	8	16
$iter$	124	1022	377	601	1554	$2000^{\sharp}$
nf	129	2212	804	1278	3360	4322
ng	124	1106	402	639	1680	2161
f_f	4.62_{-12}	8.18_{-12}	4.50_{-11}	4.64_{-12}	1.54_{-11}	1.27_{-9}
RMSE	6.34_{-6}	1.01_{-5}	1.74_{-5}	1.46_{-5}	1.10_{-4}	2.56_{-3}
t	2.12	1.27_2	5.25_1	9.35_1	3.48_2	6.86_2

Table 1: Comparisons of Algorithm 2 and LR-FISTA for the noiseless PhaseLift problem (3.2) with $n_1 = n_2 = 64$ and several values of k. $\sharp$ represents the number of iterations reach the maximum. $iter$, nf, ng, f_f, and t denote the number of iteration, the number of function evaluation, the number of gradient evaluation, final cost function value and computational time in second respectively.

6.2 Comparisons with a Standard Low-rank Method

Candès et al., [9, 12] use a Matlab library TFOCS [6] that contains a variety of accelerated first-order methods given in [21] and, in particular, the method based on FISTA [5] is used to optimize the cost functions in PhaseLift. For large-scale problems, a low rank version of FISTA called LR-FISTA is used. The main difference is that the iterates of LR-FISTA are low-rank matrices computed via projection rather than the full-rank iterates of FISTA.

As in [9], the difference between the true solution and the minimizer is measured by the relative mean-square error (RMSE) $\min_{a:|a|=1} \|ax - x_*\|_2 / \|x_*\|_2$ and by $10\log_{10}(\text{RMSE})$ when expressed in dB. The scale of the noise is measured by the signal-to-noise ratio (SNR) in dB given by $\text{SNR} = 10\log_{10}(\|b\|_2^2 / \|b - \hat{b}\|_2^2)$, where $b = \operatorname{diag}(Z x_* x_*^* Z^*)$ and $\hat{b}$ is the noise measurements.

Tables 1 and 2 report experimental results for Algorithm 2 and LR-FISTA for the noiseless and Gaussian noise problems (3.2) and (3.4) respectively. For the Gaussian noise problem, τ is 10^{-4} and the corresponding SNR is 31.05 dB in this experiment. Multiple examples with different random seeds and different SNR show similar results. First, increasing k for LR-FISTA usually does not improve the performance in the sense of efficiency and effectiveness for both noiseless and Gaussian noise problems. Second, increasing κ usually does not reduce the RMSE. When it does, the RMSE values are not reduced significantly. Therefore, $\kappa = 0$ is used in the later comparisons for Gaussian noise problems. Third, Algorithm 2 outperforms LR-FISTA significantly in the sense that Algorithm 2 provides similar accuracy usually while requiring fewer operations of all types (cost function evaluation, gradients etc.) and yielding a significantly smaller computational time.

7 Conclusion

In this paper, the recently proposed PhaseLift framework for solving the phase retrieval problem has motivated the consideration of a class of cost functions on the set of complex Hermitian positive semidefinite matrices $\mathcal{D}_n$. An alternate cost function F related to factorization is used to replace any cost function H in this class. The important optimality condition, Theorem 4.1, shows that if Y_p is a rank deficient minimizer of F_p, then $Y_p Y_p^*$ is a stationary point of H. Additionally, Algorithm 2 based on optimization on a fixed rank manifold and dynamically reducing rank is developed for optimizing the cost function F. For optimization on a fixed rank manifold,

noise	κ	Algorithm 2	LR-FISTA 1	2	4	8	16
$iter$	10^{-6}	128	1027	$2000^\sharp$	$2000^\sharp$	$2000^\sharp$	$2000^\sharp$
	0	138	1070	$2000^\sharp$	$2000^\sharp$	$2000^\sharp$	$2000^\sharp$
nf	10^{-6}	132	2210	4266	4312	4336	4316
	0	143	2306	4308	4322	4314	4320
ng	10^{-6}	128	1105	2712	2156	2168	2158
	0	138	1153	2154	2161	2157	2160
f_f	10^{-6}	1.84_{-5}	1.84_{-5}	1.91_{-5}	2.35_{-5}	3.55_{-5}	7.62_{-5}
	0	4.08_{-7}	4.08_{-7}	1.16_{-6}	6.27_{-6}	2.51_{-5}	8.89_{-5}
RMSE	10^{-6}	6.72_{-4}	6.72_{-4}	1.09_{-3}	2.10_{-3}	3.53_{-3}	6.27_{-3}
	0	6.70_{-4}	6.70_{-4}	1.09_{-3}	2.18_{-3}	4.01_{-3}	7.29_{-3}
t	10^{-6}	2.13	1.27_2	2.75_2	3.01_2	4.64_2	7.04_2
	0	2.20	1.34_2	2.63_2	2.98_2	4.32_2	6.91_2

Table 2: Comparisons of Algorithm 2 and LR-FISTA for the noise PhaseLift problem (3.4) with SNR be 31.05 dB, $n_1 = n_2 = 64$ and several values of k and κ. $\sharp$ represents the number of iterations reach the maximum.

recently developed state-of-the-art Riemannian optimization methods on a quotient space are used. In the experiments, it is shown that Algorithm 2 with LRBFGS yields more efficient than LR-FISTA for both noiseless and noise artificial data. The code used for these experiments is available at `http://www.math.fsu.edu/~whuang2/papers/SPLRROMCSCshort.htm`.

References

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[2] P.-A. Absil, M. Ishteva, L. De Lathauwer, and S. Van Huffel. A geometric Newton method for Oja's vector field. *Neural Computationomputation*, 21(5):1415–33, May 2009. doi:10.1162/neco.2008.04-08-749.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.

[4] C. G. Baker. *Riemannian manifold trust-region methods with applications to eigenproblems*. PhD thesis, Florida State University, Department of Computational Science, 2008.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009. doi:10.1137/080716542.

[6] S. Becker, E. J. Cand, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3:165–218, 2011.

[7] O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D. K. Satapathy, and J. F. van der Veen. Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels. *Acta crystallographica. Section A, Foundations of crystallography*, 63(Pt 4):306–314, July 2007. doi:10.1107/S0108767307021903.

[8] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, February 2003. doi:10.1007/s10107-002-0352-8.

[9] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225, 2013. arXiv:1109.0573v2.

[10] E. J. Candès and X. Li. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, June 2013. doi:10.1007/s10208-013-9162-z.

[11] E. J. Candés, X. Li, and M. Soltanolkotabi. Phase retrieval via Wirtinger flow: theory and algorithms. *IEEE Transactions on Information Theory*, 64(4):1985–2007, 2016.

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

[13] C. Chen, J. Miao, C. w. Wang, and T. K. Lee. Application of optimization technique to non-crystalline x-ray diffraction microscopy: Guided hybrid input-output method. *Physical Review B*, 76(6):064113, August 2007. doi:10.1103/PhysRevB.76.064113.

[14] R. W. Harrison. Phase problem in crystallography. *Journal of the Optical Society of America A*, 10(5):1046–1055, May 1993. doi:10.1364/JOSAA.10.001046.

[15] W. Huang. *Optimization algorithms on Riemannian manifolds with applications*. PhD thesis, Florida State University, Department of Mathematics, 2013.

[16] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Mathematical Programming*, 150(2):179–216, February 2015.

[17] Wen Huang, K. A. Gallivan, and P.-A. Absil. A Broyden Class of Quasi-Newton Methods for Riemannian Optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

[18] Wen Huang, K. A. Gallivan, and Xiangxiong Zhang. Solving phaselift by low rank riemannian optimization methods. In *Proceedings of of the International Conference on Computational Science (ICCS2016), accepted*, 2016.

[19] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[20] J. Miao, T. Ishikawa, Q. Shen, and T. Earnest. Extending X-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. *Annual review of physical chemistry*, 59:387–410, January 2008. doi:10.1146/annurev.physchem.59.032607.093642.

[21] Y. Nesterov. *Introductory lectures on convex programming: a basic course*, volume I. Springer, 2004.

[22] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, January 2012. doi:10.1137/11082885X.

[23] J. L. C. Sanz. Mathematical considerations for the problem of Fourier transform phase retrieval from magnitude. *SIAM Journal on Applied Mathematics*, 45(4):651–664, 1985.

[24] H. Sato. A Dai-Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions. *Computational Optimization and Applications*, 2015. to appear.

[25] I. Waldspurger, A. DAspremont, and S. Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, December 2013. doi:10.1007/s10107-013-0738-9.

[26] A. Walther. The question of phase retrieval in optics. *Optica Acta: International Journal of Optics*, 10(1):41–49, January 1963. doi:10.1080/713817747.