Phase Retrieval

*Project 4 on the Course "Algorithms for Big Data Analysis".

1st Chen Yihang

Peking University

1700010780

Abstract

In this report, I implemented the PhaseLift [Candes et al., 2013] and Wirtinger flow [Candes et al., 2015] to solve the phase retrieval problem. We test the two algorithms on the coded diffraction model, and find out that Wirtinger flow is much more efficient. Besides, we also discuss the "probability of successful retrieval" on different m/n.

CONTENTS

I. Problem setting

One popular formulation of the phase retrieval problem is solving a system of quadratic equations in the form

$$y_r = |\langle \boldsymbol{a}_r, \boldsymbol{z} \rangle|^2, \quad r = 1, 2, \ldots, m, \tag{I.1}$$

where $z \in \mathbb{C}^n$ is the decision variable, $a_r \in \mathbb{C}^n$ are known sampling vectors, $< a_r, z >$ is the inner product between a_r and z in $\mathbb{C}^n$, $|a|$ is the magnitude of $a \in \mathbb{C}$, and $y \in \mathbb{R}$ are the observed measurements. This problem is a general instance of a nonconvex quadratic program (QP). In this report, I implemented the gradient method of Wirtinger flow [Candes et al., 2015] and PhaseLift [Candes et al., 2013]. In the following, I give a brief introduction to both of the algorithms.

II. Wirtinger flow

The Wirtinger flow includes two steps: (1) a careful initialization obtained by means of a spectral method, and (2) a series of updates refining this initial estimate by iteratively applying a novel update rule, much like in a gradient descent scheme.

A. *Minimization of a non-convex objective*

Let $\ell(x, y)$ be a loss function measuring the misfit between both its scalar arguments. If the loss function is non-negative and vanishes only when $x = y$, then a solution to the generalized phase retrieval problem (I.1) is any solution to

$$\text{minimize} \quad f(\boldsymbol{z}) \coloneqq \frac{1}{2m} \sum_{r=1}^{m} \ell\big(y_r, |\boldsymbol{a}_r^* \boldsymbol{z}|^2\big), \quad \boldsymbol{z} \in \mathbb{C}^n. \tag{II.1}$$

Although one could study many loss functions, we shall focus in this paper on the simple quadratic loss $\ell(x, y) = (x - y)^2$. Admittedly, the formulation (II.1) does not make the problem any easier since the function f is not convex. Minimizing non-convex objectives, which may have very many stationary points, is known to be NP-hard in general. In fact, even establishing convergence to a local minimum or stationary point can be quite challenging.

Our approach to (II.1) is simply stated: start with an initialization $\boldsymbol{z}_0$, and for $\tau = 0, 1, 2, \ldots$, inductively define

$$\boldsymbol{z}_{\tau+1} = \boldsymbol{z}_\tau - \frac{\mu_{\tau+1}}{\|\boldsymbol{z}_0\|^2}\left(\frac{1}{m}\sum_{r=1}^{m}\left(|\boldsymbol{a}_r^*\boldsymbol{z}|^2 - y_r\right)(\boldsymbol{a}_r\boldsymbol{a}_r^*)\boldsymbol{z}\right) := \boldsymbol{z}_\tau - \frac{\mu_{\tau+1}}{\|\boldsymbol{z}_0\|^2}\nabla f(\boldsymbol{z}_\tau). \tag{II.2}$$

If the decision variable $\boldsymbol{z}$ and the sampling vectors were all real valued, the term between parentheses would be the gradient of f divided by two, as our notation suggests. However, since $f(\boldsymbol{z})$ is a mapping from $\mathbb{C}^n$ to $\mathbb{R}$, it is not holomorphic and hence not complex-differentiable. However, this term can still be viewed as a gradient based on Wirtinger derivatives. Hence, (II.2) is a form of steepest descent and the parameter $\mu_{\tau+1}$ can be interpreted as a step size (note nonetheless that the effective step size is also inversely proportional to the magnitude of the initial guess).

B. Initialization via a spectral method

Our main result states that for a certain random model, if the initialization $\boldsymbol{z}_0$ is sufficiently accurate, then the sequence $\{\boldsymbol{z}_\tau\}$ will converge toward a solution to the generalized phase problem (I.1). In this paper, we propose computing the initial guess $\boldsymbol{z}_0$ via a spectral method, detailed in Algorithm 1. In words, $\boldsymbol{z}_0$ is the leading eigenvector of the positive semidefinite Hermitian matrix $\sum_r y_r \boldsymbol{a}_r\boldsymbol{a}_r^*$ constructed from the knowledge of the sampling vectors and observations. (As usual, $\boldsymbol{a}_r^*$ is the adjoint of $\boldsymbol{a}_r$.) Letting $\boldsymbol{A}$ be the $m \times n$ matrix whose rth row is $\boldsymbol{a}_r^*$ so that with obvious notation $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$, $\boldsymbol{z}_0$ is the leading eigenvector of $\boldsymbol{A}^* \operatorname{diag}\{\boldsymbol{y}\}\, \boldsymbol{A}$ and can be computed via the power method by repeatedly applying $\boldsymbol{A}$, entrywise multiplication by $\boldsymbol{y}$ and $\boldsymbol{A}^*$.

We consider a simple model where everything is real valued, and in which the vectors a_r are i.i.d. $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Also without any loss in generality and to simplify exposition in this section we shall assume $\|\boldsymbol{x}\| = 1$.

Let $\boldsymbol{x}$ be a solution to (I.1) so that $y_r = |\langle \boldsymbol{a}_r, \boldsymbol{x}\rangle|^2$, and consider the initialization step first. In the Gaussian model, a simple moment calculation gives

$$\mathbb{E}\left[\frac{1}{m}\sum_{r=1}^{m} y_r \boldsymbol{a}_r\boldsymbol{a}_r^*\right] = \boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*.$$

Algorithm 1 Wirtinger Flow: Initialization

Input: Observations $\{y_r\} \in \mathbb{R}^m$.

Set

$$\lambda^2 = n \frac{\sum_r y_r}{\sum_r \|\boldsymbol{a}_r\|^2}.$$

Set $\boldsymbol{z}_0$, normalized to $\|\boldsymbol{z}_0\| = \lambda$, to be the eigenvector corresponding to the largest eigenvalue of

$$\boldsymbol{Y} = \frac{1}{m} \sum_{r=1}^{m} y_r \boldsymbol{a}_r \boldsymbol{a}_r^*.$$

Output: Initial guess $\boldsymbol{z}_0$.

By the strong law of large numbers, the matrix $\boldsymbol{Y}$ in Algorithm 1 is equal to the right-hand side in the limit of large samples. Since any leading eigenvector of $\boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ is of the form $\lambda \boldsymbol{x}$ for some scalar $\lambda \in \mathbb{R}$, we see that the intialization step would recover $\boldsymbol{x}$ perfectly, up to a global sign or phase factor, had we infinitely many samples. Indeed, the chosen normalization would guarantee that the recovered signal is of the form $\pm\boldsymbol{x}$. As an aside, we would like to note that the top two eigenvalues of $\boldsymbol{I} + 2\boldsymbol{x}\boldsymbol{x}^*$ are well separated unless $\|\boldsymbol{x}\|$ is very small, and that their ratio is equal to $1 + 2\|\boldsymbol{x}\|^2$. Now with a finite amount of data, the leading eigenvector of $\boldsymbol{Y}$ will of course not be perfectly correlated with $\boldsymbol{x}$ but we hope that it is sufficiently correlated to point us in the right direction.

III. PhaseLift Algorithm

A. Lifting

Suppose we have $x_0 \in \mathbb{C}^n$ or $\mathbb{C}^{n_1 \times n_2}$ (or some higher-dimensional version) about which we have quadratic measurements of the form

$$\mathbb{A}(x_0) = \{|\langle a_k, x_0\rangle|^2 : k = 1, 2, \ldots, m\}. \tag{III.1}$$

Phase retrieval is then the feasibility problem

$$\begin{array}{ll} \text{find} & x \\ \text{obeying} & \mathbb{A}(x) = \mathbb{A}(x_0) := b. \end{array} \tag{III.2}$$

As is well known, quadratic measurements can be lifted up and interpreted as linear measurements about the rank-one matrix $X = xx^*$. Indeed,

$$|\langle a_k, x\rangle|^2 = \mathrm{Tr}(x^* a_k a_k^* x) = \mathrm{Tr}(a_k a_k^* x x^*) := \mathrm{Tr}(A_k X),$$

where A_k is the rank-one matrix $a_k a_k^*$. In what follows, we will let $\mathcal{A}$ be the linear operator mapping positive semidefinite matrices into $\{\mathrm{Tr}(A_k X) : k = 1, \ldots, m\}$. Hence, the phase retrieval problem is equivalent to

$$\begin{array}{ll} \text{find} & X \\ \text{subject to} & \mathcal{A}(X) = b \\ & X \succeq 0 \\ & \mathrm{rank}(X) = 1 \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \text{minimize} & \mathrm{rank}(X) \\ \text{subject to} & \mathcal{A}(X) = b \\ & X \succeq 0. \end{array} \tag{III.3}$$

Upon solving the left-hand side of (III.3), we would factorize the rank-one solution X as xx^*, hence finding solutions to the phase-retrieval problem. Note that the equivalence between the left- and right-hand side of (III.3) is straightforward since by definition, there exists a rank-one solution. Therefore, our problem is a rank minimization problem over an affine slice of the positive semidefinite cone. As such, it falls in the realm of low-rank *matrix completion* or *matrix recovery*, a class of optimization problems that has gained tremendous attention in recent years. Just as in matrix completion, the linear system $\mathcal{A}(X) = b$, with unknown in the positive semidefinite cone, is highly underdetermined. For instance suppose our signal x_0 has n complex unknowns. Then we may imagine collecting six diffraction patterns with n measurements for each (no oversampling). Thus $m = 6n$ whereas the dimension of the space of $n \times n$ Hermitian matrices over the reals is n^2, which is obviously much larger.

B. Recovery via convex programming

The rank minimization problem (III.3) is NP hard. We propose using the trace norm as a convex surrogate for the rank functional, giving the familiar SDP (and a crucial component of PhaseLift),

$$\begin{array}{ll} \text{minimize} & \text{trace}(X) \\ \text{subject to} & \mathcal{A}(X) = b \\ & X \succeq 0. \end{array} \tag{III.4}$$

In our implementation, we propose to solve the following convex optimization problem in CVX. Here, λ can be set to zero.

$$\begin{array}{ll} \text{minimize} & \|\mathcal{A}(X) - b\| + \lambda \|X\|_1 \\ & X \succeq 0. \end{array} \tag{III.5}$$

IV. Numerical Experiments

We present some numerical experiments to assess the empirical performance of the Wirtinger flow algorithm. Here, we mostly consider a model of coded diffraction patterns reviewed below.

A. The coded diffraction model

We consider an acquisition model, where we collect data of the form

$$y_r = \left| \sum_{t=0}^{n-1} x[t] \bar{d}_\ell(t) e^{-i2\pi kt/n} \right|^2, \quad r = (\ell, k), \quad \begin{array}{c} 0 \le k \le n-1 \\ 1 \le \ell \le L \end{array}; \tag{IV.1}$$

thus for a fixed ℓ, we collect the magnitude of the diffraction pattern of the signal $\{x(t)\}$ modulated by the waveform/code $\{d_\ell(t)\}$. By varying ℓ and changing the modulation pattern d_ℓ, we generate several views thereby creating a series of *coded diffraction patterns* (CDPs).

In this report, we are mostly interested in the situation where the modulation patterns are random; in particular, we study a model in which the d_ℓ's are i.i.d. distributed, each having i.i.d. entries sampled from a distribution d.

$$\mathbb{E}\, d = 0, \quad \mathbb{E}\, d^2 = 0, \quad \mathbb{E}\, |d|^4 = 2(\mathbb{E}\, |d|^2)^2. \tag{IV.2}$$

A random variable obeying these assumptions is said to be *admissible*. Since d is complex valued we can have $\mathbb{E}\, d^2 = 0$ while $d \neq 0$. An example of an admissible random variable is $d = b_1 b_2$, where b_1 and b_2 are independent and distributed as

$$b_1 = \begin{cases} +1 & \text{with prob. } 1/4 \\ -1 & \text{with prob. } 1/4 \\ -i & \text{with prob. } 1/4 \\ +i & \text{with prob. } 1/4 \end{cases} \quad \text{and} \quad b_2 = \begin{cases} \sqrt{2}/2 & \text{with prob. } 4/5 \\ \sqrt{3} & \text{with prob. } 1/5 \end{cases}. \tag{IV.3}$$

We shall refer to this distribution as an *octanary* pattern since d can take on eight distinct values. The condition $\mathbb{E}[d^2] = 0$ is here to avoid unnecessarily complicated calculations in our theoretical analysis. In particular, we can also work with a *ternary* pattern in which d is distributed as

$$d = \begin{cases} +1 & \text{with prob. } 1/4 \\ 0 & \text{with prob. } 1/2 \\ -1 & \text{with prob. } 1/4 \end{cases}. \tag{IV.4}$$

B. The Gaussian and coded diffraction models

We begin by examining the performance of the Wirtinger flow algorithm for recovering random signals $\boldsymbol{x} \in \mathbb{C}^n$ under the Gaussian and coded diffraction models. We are interested in signals of two different types:

- *Random low-pass signals.* Here, $\boldsymbol{x}$ is given by

$$x[t] = \sum_{k=-(M/2-1)}^{M/2} (X_k + iY_k) e^{2\pi i (k-1)(t-1)/n},$$

  with $M = n/8$ and X_k and Y_k are i.i.d. $\mathcal{N}(0,1)$.
- *Random Gaussian signals.* In this model, $\boldsymbol{x} \in \mathbb{C}^n$ is a random complex Gaussian vector

with i.i.d. entries of the form $x[t] = X + iY$ with X and Y distributed as $\mathcal{N}(0,1)$; this can be expressed as

$$x[t] = \sum_{k=-(n/2-1)}^{n/2} (X_k + iY_k) e^{2\pi i(k-1)(t-1)/n},$$

where X_k and Y_k are are i.i.d. $\mathcal{N}(0,1/8)$ so that the low-pass model is a 'bandlimited' version of this high-pass random model (variances are adjusted so that the expected signal power is the same).

Below, we set $n = 128$, and generate one signal of each type which will be used in all the experiments.

The initialization step of the Wirtinger flow algorithm is run by applying 50 iterations of the power method. In the iteration (II.2), we use the parameter value $\mu_\tau = \min(1 - \exp(-\tau/\tau_0), 0.2)$ where $\tau_0 \approx 330$. We stop after $2,500$ iterations, and report the empirical probability of success for the two different signal models. The empirical probability of succcess is an average over 100 trials, where in each instance, we generate new random sampling vectors according to the Gaussian or CDP models. We declare a trial successful if the relative error of the reconstruction $\mathrm{dist}(\hat{\boldsymbol{x}}, \boldsymbol{x})/\left\|\boldsymbol{x}\right\|$ falls below 10^{-5}, where we define $dist(\boldsymbol{z}, \boldsymbol{x}) = \min_{\phi\in[0,2\pi]} \ \left\|\boldsymbol{z} - e^{i\phi}\boldsymbol{x}\right\|$.

C. *Results presentation*

Wirtinger Flow

In this part of the experiments, we set $n = 32, 64, 128, 256$.

In order to investigte the best choice of $L = m/n$, we perform a grid search, and evaluate 100 times to estimate the probability of success. A retrieval is called successful if $\|y - |Az_k|^2\|_2/\|y\| < 10^{-5}$, where z_k is the retrieval vector. We use MATLAB file "test_L.m" to generate the Figure 1, 2, 3, and 4. We find that for different n, similar Ls are reuqired to guarantee reconstruction.

Then, we set $n = 128$. In the Gaussian model, we use $L = 4.5$ and in the coded diffusion model, we use $L = 6$, which is presented in Figure 5.

PhaseLift

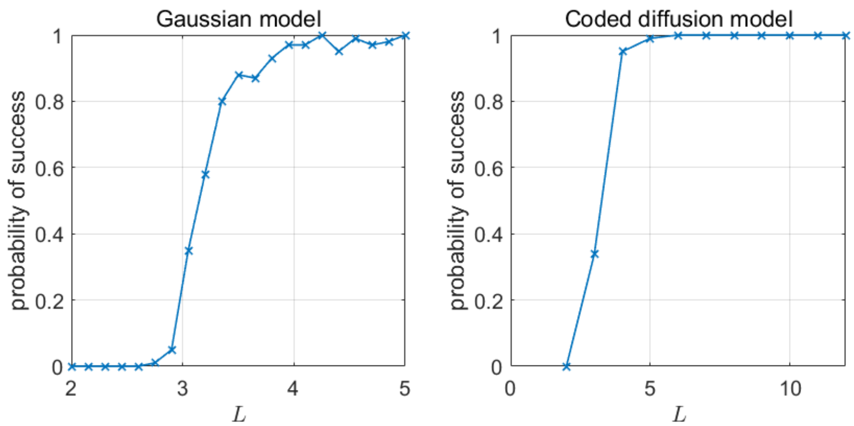


Fig. 1: Probability of success under different L, $n = 32$

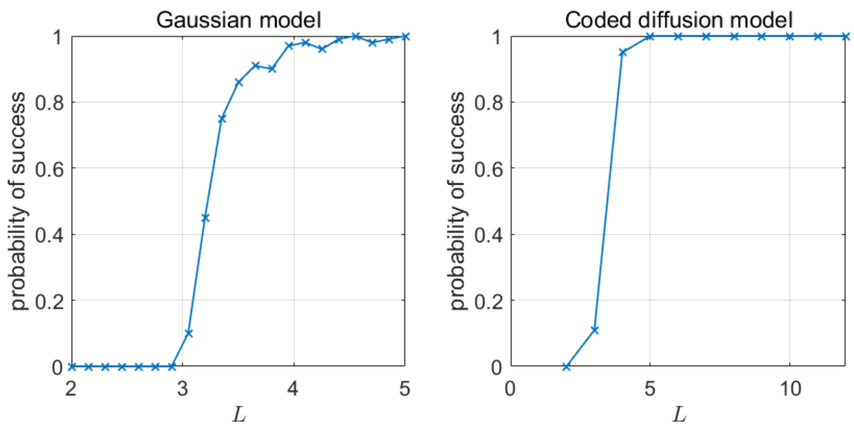


Fig. 2: Probability of success under different L, $n = 64$

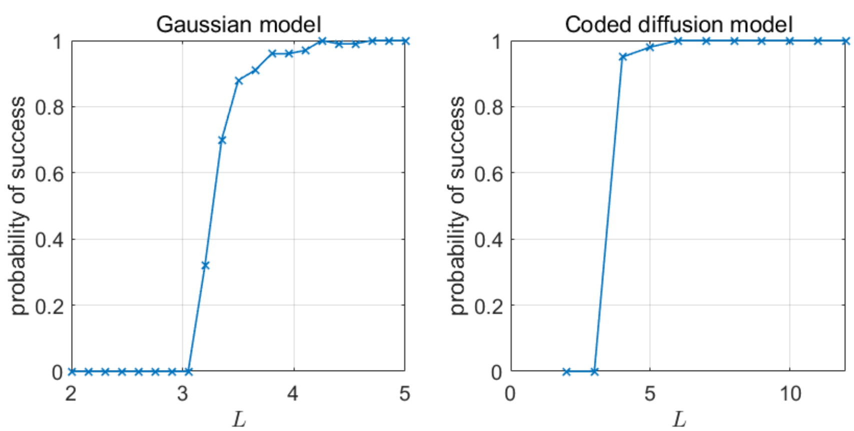


Fig. 3: Probability of success under different L, $n = 128$

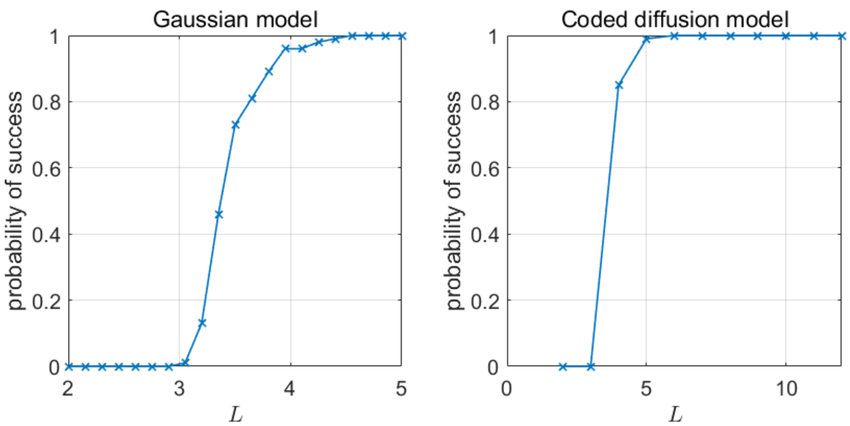
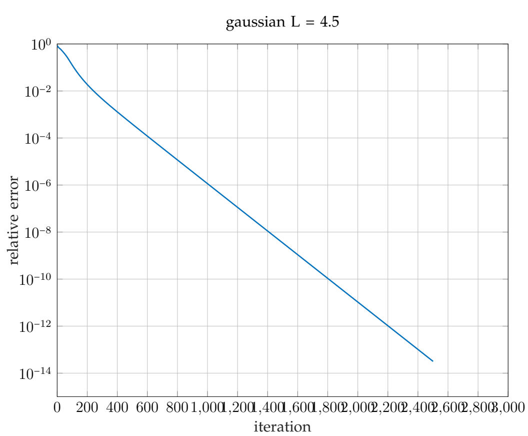
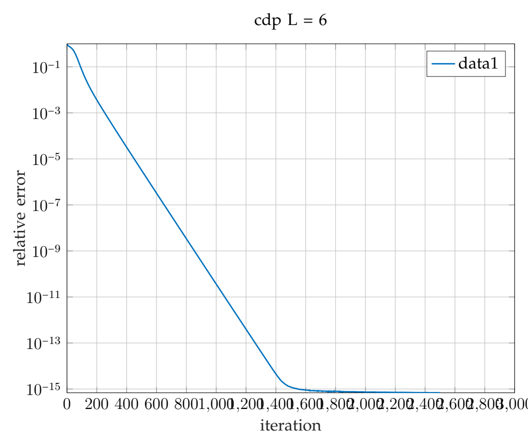


Fig. 4: Probability of success under different L, $n = 256$



(a) Gaussian model



(b) Coded diffusion models

Fig. 5: Relative error-iteration

We find that CVX-based PhaseLift cannot solve large-scale problem when $n = 128$. Hence, we only test its performance on small scale problems. However, Wirtinger flow still significantly outperforms the CVX-based PhaseLift.

We take $n = 20$ for example, and L is set to be 6 in the CDF case. The results can be summarized below, the error is defined by $\text{dist}(\hat{\boldsymbol{x}}, \boldsymbol{x}) / \|\boldsymbol{x}\|$

References

[Candes et al., 2013] Candes, E. J., Eldar, Y. C., Strohmer, T., and Voroninski, V. (2013). Phase retrieval via matrix completion. *Siam Journal on Imaging Sciences*, 6(1):199–225.

algorithm	time	error
Wirtinger flow	0.1276	2.7944e-16
CVX-based PhaseLift	0.9412	1.7403e-11

[Candes et al., 2015] Candes, E. J., Li, X., and Soltanolkotabi, M. (2015). Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007.