Recovering Data from Underdetermined Quadratic Measurements (CS 229a Project: Final Writeup)

Mahdi Soltanolkotabi

December 16, 2011

1 Introduction

Data that arises from engineering applications often contains some type of low dimensional structure that enables intelligent representation and processing. That is, the data is on an unknown low dimensional structure in a very high dimensional space. However, in many engineering disciplines we do not have access to the data directly, but only have access to measurements of the data. In some of these engineering disciplines we would like to get as few measurements as possible, while preserving quality (a good example is MR imaging where the number of measurements corresponds to MR scan time). In others, the numbers of measurements are limited by the nature of the problem or some physical constraint (e.g. the Netflix challenge where we only have access to only a few user ratings). Therefore in many practical settings we face a very challenging problem: recovering data from underdetermined measurements.

The past decade has witnessed a tremendous breakthrough in the recovery of low-dimensional data (sparse/low rank) from underdetermined *linear* measurements (e.g. compressed sensing/matrix completion). However in many engineering applications the measurements are not linear, one such example is phase retrieval.

Phase retrieval is the problem of recovering a general signal, e.g. an image from the magnitude of its Fourier transform (equivalent to quadratic measurement of the signal). Phase retrieval has numerous applications: X-ray crystallography, imaging science, diffraction imaging, optics, astronomical imaging, and microscopy, to name just a few.

A number of methods have been developed to address the phase retrieval problem over the last century. There are two classes of methods for the phase retrieval problem: The first class are methods that are algebraic in nature and thus not robust to noise. The second, class are methods that rely on non-convex optimization schemes and often get stuck in local minima. In this class there has been heavy use of machine learning type algorithms in the literature e.g. Neural Networks, graphical modeling, etc. Very recently new methods have been developed (based on SDP relaxations), that are both robust to noise and are convex in nature.

Despite these advances these new methods do not utilize the low dimensional structure inherent in many phase retrieval applications (e.g. sparsity of images in a given frame). Using these inherent low dimensional structures of image signals one might be able to dramatically decrease the number of measurements needed for many phase retrieval applications.

In this paper we present a unified view of how data can be recovered from quadratic measurements. We show that the aforementioned SDP relaxation in the context of phase retrieval can be derived as a special case of this more general formulation. We will show that our general formulation leads to new algorithms and insights in noisy phase retrieval. We also consider the case where the quadratic measurements are underdetermined i.e. there is no unique solution. In this case, we will see that one can still recover the data using apriori information in the form of sparsity of data in a dictionary. We will show that this formulation leads to a new algorithm (Sparse Phase Lift) for the phase retrieval problem, that utilizes the inherent low dimensional structure of the data to decrease the number of measurements needed in phase retrieval.

2 Unsupervised Learning of data from Quadratic Measurements

In the first part of this section we show that one can use Shor's Semidefinite Relaxation scheme [2], to recover a vector from quadratic measurements as a convex optimization problem. In the second part we study the underdetermined case and present a convex optimization formulation for recovery of data from underdetermined quadratic measurements.

2.1 General Case

Assume that we wish to find a data vector $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies a set of m quadratic equations. We formulate this as a convex optimization problem by adding a quadratic objective function.[1]

$$\begin{aligned} \min f_0(\boldsymbol{x}) = \quad & \boldsymbol{x}^T \boldsymbol{A}_0 \boldsymbol{x} + 2\boldsymbol{b}_0^T \boldsymbol{x} + c_0 \\ y_1 = \quad & \boldsymbol{x}^T \boldsymbol{A}_1 \boldsymbol{x} + 2\boldsymbol{b}_1^T \boldsymbol{x} + c_1 \\ \vdots \quad & \\ y_m = \quad & \boldsymbol{x}^T \boldsymbol{A}_m \boldsymbol{x} + 2\boldsymbol{b}_m^T \boldsymbol{x} + c_m \end{aligned}$$

Notice that this primal problem is non-convex. To bound from bellow the optimal value we build the dual problem.

$$\begin{aligned} f_\lambda(\boldsymbol{x}) = \quad & f_0(\boldsymbol{x}) + \sum_{i=1}^m \lambda_i f_i(\boldsymbol{x}) \\ = \quad & \boldsymbol{x}^T \boldsymbol{A}(\lambda) \boldsymbol{x} + 2\boldsymbol{b}^T(\lambda) \boldsymbol{x} + c(\lambda) \end{aligned}$$

where

$$\boldsymbol{A}(\lambda) = \boldsymbol{A}_0 + \sum_{i=1}^m \lambda_i \boldsymbol{A}_i; \boldsymbol{b}(\lambda) = \boldsymbol{b}_0 + \sum_{i=1}^m \lambda_i \boldsymbol{b}_i; c(\lambda) = c_0 + \sum_{i=1}^m \lambda_i c_i.$$

Notice that $\min_{\boldsymbol{x} \in \mathbb{R}^n} f_\lambda(\boldsymbol{x})$ is a lower bound for the original problem. Now the method we are going to use in order to relax (2.1) is by finding the best lower bound of $f_\lambda(\boldsymbol{x}) \geq \eta$, by an appropriate convex search over all λ and η. Now notice that

$$\boldsymbol{x}^T \boldsymbol{A}(\lambda) \boldsymbol{x} + 2\boldsymbol{b}^T(\lambda) \boldsymbol{x} + c(\lambda) - \eta \geq 0,$$

if and only if $(\lambda, \eta) \in \mathbb{R}^m \times \mathbb{R}$ satisfy

$$\begin{bmatrix} c(\lambda) & \boldsymbol{b}(\lambda)^T \\ \boldsymbol{b}(\lambda)^T & \boldsymbol{A}(\lambda) \end{bmatrix} \succeq 0.$$

[1] Notice that if one is only interested in recovery from quadratic equality constraints one could set the objective to 0, i.e. $\boldsymbol{A}_0 = \boldsymbol{0}$, $\boldsymbol{b}_0 = \boldsymbol{0}$, and $c_0 = 0$. We keep the general form because it will prove useful in deriving results in noisy phase retrieval.

So inorder to find a good lower bound we maximize the above expression, i.e.

$$\max_{\eta,\lambda}\left\{\eta : \begin{bmatrix} c_0 + \sum_{i=1}^m \lambda_i c_i - \eta & \boldsymbol{b}_0^T + \sum_{i=1}^m \lambda_i \boldsymbol{b}_i^T \\ \boldsymbol{b}_0 + \sum_{i=1}^m \lambda_i \boldsymbol{b}_i & \boldsymbol{A}_0 + \sum_{i=1}^m \lambda_i \boldsymbol{A}_i \end{bmatrix} \succeq 0\right\}.$$

If we write the semidefinite dual of the above problem we get

$$\begin{aligned} &\min_{\boldsymbol{X}\in\mathbb{R}^{(n+1)\times(n+1)}} Tr(\boldsymbol{A}_0\boldsymbol{X}) \\ &\text{subject to} \\ &\mathrm{Tr}\left(\begin{bmatrix} c_i & \boldsymbol{b}_i^T \\ \boldsymbol{b}_i & \boldsymbol{A}_i \end{bmatrix}\boldsymbol{X}\right) = y_i, i = 1,\dots,m \\ &\boldsymbol{X} \succeq 0; \boldsymbol{X}_{11} = 1 \end{aligned}$$

where we have excluded the details of the last derivation due to space constraints.

2.2 Underdetermined Case

Assume that the aforementioned quadratic equality constraints are underdetermined, i.e. even if one is able to use combinatorial search, there is not a unique solution to the aforementioned quadratic equations (In fact in many problems of interest there are exponentially many such solutions). Furthermore, assume we have some additional information about the data vector $\boldsymbol{x}$ in the form of sparsity in a given dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times N}$ with $N \geq n$; i.e. $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{s}$, where s is a sparse vector. We will assume that $\boldsymbol{D}$ is a frame and therefore $\boldsymbol{D}\boldsymbol{D}^T = \boldsymbol{I}$. In this case we have

$$\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^T = \boldsymbol{D}\boldsymbol{s}\boldsymbol{s}^T\boldsymbol{D}^T,$$

and using the fact that $\boldsymbol{D}$ is a frame,

$$\boldsymbol{D}^T\boldsymbol{X}\boldsymbol{D} = \boldsymbol{s}\boldsymbol{s}^T.$$

This implies that if $\boldsymbol{s}$ is sparse, then so is $\boldsymbol{D}^T\boldsymbol{X}\boldsymbol{D}$. Based on this observation we add a regularization term to (2.2) to promote sparsity. We do this by adding the convex surrogate of sparsity which is the ℓ_1 norm. Thus, we arrive at the following optimization formulation for recovering data from underdetermined quadratic measurements,

$$\begin{aligned} &\min_{\boldsymbol{X}\in\mathbb{R}^{(n+1)\times(n+1)}} Tr(\boldsymbol{A}_0\boldsymbol{X}) + \lambda\left\|\boldsymbol{D}\boldsymbol{X}\boldsymbol{D}^T\right\|_{\ell_1} \\ &\text{subject to} \\ &\mathrm{Tr}\left(\begin{bmatrix} c_i & \boldsymbol{b}_i^T \\ \boldsymbol{b}_i & \boldsymbol{A}_i \end{bmatrix}\boldsymbol{X}\right) = y_i, i = 1,\dots,m \\ &\boldsymbol{X} \succeq 0; \boldsymbol{X}_{11} = 1. \end{aligned}$$

3 Phase Retrieval

3.1 Basic Formulation

Phase retrieval is about recovering a general signal, from the magnitude of its Fourier transform. This problem arises because detectors can often only record the squared modulus of the Fresnel or Fraunhofer diffraction pattern of the radiation that is scattered from an object. Therefore, the phase of the signal is lost and one faces the very challenging problem of recovering the phase of a signal given magnitude measurements. We now give a 1-D formulation of the problem using finite length signals. Suppose we have a 1D signal $\boldsymbol{x} = (\boldsymbol{x}[0], \boldsymbol{x}[1], \dots, \boldsymbol{x}[n-1]) \in \mathbb{C}^n$ and write its Fourier transform

$$\hat{\boldsymbol{x}}[\omega] = \frac{1}{\sqrt{n}} \sum_{0\leq t<n} \boldsymbol{x}[t]e^{-i2\pi\omega t/n},\ \omega \in \Omega.$$

where Ω is a grid of sampled frequencies. The phase retrieval problem is finding $\boldsymbol{x}$ from the magnitude coefficients $|\hat{\boldsymbol{x}}[\omega]|$. Before going further one should notice that there are some inherent ambiguities in phase retrieval in this basic form:

Figure 1: President Obama

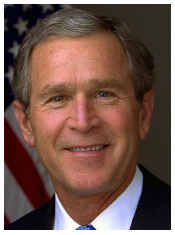


Figure 2: President Bush

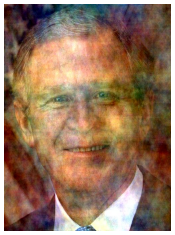


Figure 3: Magnitude of obama, phase of bush.

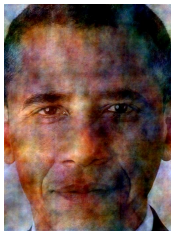


Figure 4: magnitude of bush, phase of obama.

(1) For any $\boldsymbol{a} \in \mathbb{C}$ with $|\boldsymbol{a}| = 1$, $\boldsymbol{x}$ and $c\boldsymbol{x}$ have the same magnitude.

(2) The time reversed form of the signal also has the same magnitude.

(3) the shifted version of the signal in time also has the same magnitude.

To avoid some of these ambiguities in phase retrieval applications one usually measures the magnitude of diffracted patterns (rather than just the DFT of the signal). i.e. one measures

$$y_i = |\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle|^2 : i = 1, 2, \dots, m.$$

where the diffracting waveform $\boldsymbol{a}_i[t]$ can be written as

$$\boldsymbol{a}_i[t] \propto w[t]e^{i2\pi\langle w_k, t\rangle}.$$

Therefore the goal is to recover $\boldsymbol{x}$ from measurements of the form

$$y_i = |\langle \boldsymbol{a}, \boldsymbol{x}\rangle|^2 \ \ i = 1, 2, \dots, m.$$

The careful reader will notice that there is still one ambiguity left, even in this generalized version, i.e. one cannot recover the global phase of $\boldsymbol{x}$. Therefore the goal of phase retrieval is to recover up to this global phase ambiguity.

3.2 The challenge

To emphasize the challenging aspect of this problem consider Figures 1 and 2 which contains the portraits of the two recent presidents of the U.S. Now consider Figure 3, this picture is obtained by combining the magnitude of the Obama picture and the phase of the Bush picture. As can be seen the dominant picture here is that of Bush. Figure 4 is also similar. It contains the magnitude of Bush along with the phase of Obama. These figures highlight the important role that the phase of a signal plays. Therefore if the phase of the figure (the component that seems to carry a lot information) is lost, retracting the phase only from magnitude measurements seems to be a very challenging problem.

3.3 PhaseLift

Inspired by the matrix completion problem, in a breakthrough result [3], Candes et. al. present a novel technique for the phase retrieval problem up to global phase. The convex relaxation they propose in the presence of noise is depicted in Algorithm 1.

The algorithm offers a more general framework for different noisy scenarios. What is remarkable about the result of [3] is that they

Algorithm 1 Phase Lift [3].

Input: Magnitude measurements $y_1, y_2, \ldots, y_m$ and regularization term λ.

1. Solve

$$\begin{array}{ll} \min\limits_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} & \sum_{i=1}^{m} \frac{1}{2\sigma_i^2}(y_i-\mu_i)^2 + \lambda Tr(\boldsymbol{X}) \\ \text{subject to} & \mu_i = \boldsymbol{a}_i^T \boldsymbol{X} \boldsymbol{a}_i,\ i=1,2,\ldots,m. \\ & \boldsymbol{X} \succeq 0. \end{array}$$

2. $\hat{\boldsymbol{x}} = \sigma_1 \boldsymbol{v}_1$, where σ_1 is the largest eigenvalue of the solution to the above optimization problem ($\hat{\boldsymbol{X}}$) and $\boldsymbol{v}_1$ is the corresponding eigenvector.

Output: Signal $\hat{\boldsymbol{x}}$ up to a global phase.

prove in the noise less case and when the measurement vectors $\boldsymbol{a}_i$ are gaussian random vectors (either real or complex) and as long as the number of measurements exceeds $m > cn\log n$, then $\hat{\boldsymbol{x}} = \boldsymbol{x}$ up to the global phase ambiguity we described earlier on!

3.4 Derivation of PhaseLift as a special case of the general Formulation

Now we will show that one can get to the noiseless Phase Lift algorithm using the general relaxation scheme we presented earlier. Notice that in the noiseless case the measurements are

$$y_i = |\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle| \Rightarrow y_i^2 = \boldsymbol{x}^T \boldsymbol{a}_i \boldsymbol{a}_i^T \boldsymbol{x} \tag{3.1}$$

Notice that the above equation is of the form of the general formulation, and therefore the convex relaxation in this case reduces to

$$\begin{array}{ll} \min\limits_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} & 0 \\ \text{subject to} & y_i = \boldsymbol{a}_i^T \boldsymbol{X} \boldsymbol{a}_i,\ i=1,2,\ldots,m. \\ & \boldsymbol{X} \succeq 0. \end{array} \tag{3.2}$$

Notice that this is exactly the phase lift formulation if one disregards the trace (In face the trace in [3] is only inspired by matrix completion and as the authors note themselves is not necessary). Therefore we can recover Phase Lift as a special case of our general framework.

3.5 A new Relaxation for Noisy input Phase Retrieval

The noise model considered in [3] and all of its generalizations are based on the assumption that $y_i = \mu_i + z_i$ where z_i are i.i.d. gaussian and $\mu_i = |\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle|^2$. Depending on the likelihood of z_i, then one minimized an appropriate cost function. For example the version of phase lift shown above denotes the case when z_i is i.i.d. $\mathcal{N}(0,\sigma_i^2)$. However, a more realistic noise model is

$$y_i = |\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle + z_i|^2 \tag{3.3}$$

i.e. the noise term is added to the measurements before the magnitude is taken. This form of noise amongst other things models the non-linearity in the input. While phase Lift does not offer any guidance in this case the general convex relaxation scheme we presented earlier provides a natural guidance. For example consider the case that z_i are i.i.d. $\mathcal{N}(0,\sigma_i^2)$. Notice that

$$y_i = \boldsymbol{x}^T \boldsymbol{a}_i \boldsymbol{a}_i^T \boldsymbol{x} + z_i^2 + 2z_i\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{z} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{a}_i\boldsymbol{a}_i^T & \boldsymbol{B}_i \\ \boldsymbol{B}_i^T & \boldsymbol{I}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{B}_i$ is a matrix with all entries zero except the i'th column which is $\boldsymbol{a}_i$. Also, $\boldsymbol{I}_i$ is a diagonal matrix with all entries 0 except the entry (i,i) which is one. Therefore, the maximum likelihood estimator for the gaussian noise model results in

$$\min_{\boldsymbol{x},\boldsymbol{z}\in\mathbb{R}^n} \sum_{i=1}^{m} \frac{z_i^2}{2\sigma_i^2}$$
$$\text{subject to } y_i = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{z} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{a}_i\boldsymbol{a}_i^T & \boldsymbol{B}_i \\ \boldsymbol{B}_i^T & \boldsymbol{I}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix}.$$

Notice that this is of the form of the general formulation (minimize a quadratic subject to quadratic equality constraints), and therefore one can use the general convex relaxation scheme to handle this case as well. Due to space limitations we will not write the final convex relaxation here.

4 Proposed Method (Sparse Phase Lift)

As mentioned earlier Phase Lift boasts many advantages, the optimization scheme is convex and therefore has a global solution, it is guaranteed to recover the original signal up to a global phase, and it is also robust to certain kind of noise. However, it has two major flaws (1) the optimization scheme is slow; (2) It does not take advantage of the inherent structures in many practical applications. We will address the computational cost in the next section.

Here we present an algorithm that takes advantage of the inherent low dimensionality of signals. We will call this algorithm Sparse Phase Lift. This algorithm utilizes the inherent sparsity that exists in many real world signals of interest e.g. images. So the basic assumption is that the input to Sparse Phase Lift is s-sparse with respect to a dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times N}$. Given the low dimensional nature of the signal one expects to be able to recover the signal from much lower number of measurements. We will demonstrate that Sparse Phase Lift is capable of recovering the signal with much lower number of measurements. The main steps of our suggested Sparse Phase Lift algorithm is depicted in 2.

Algorithm 2 Sparse Phase Lift(SPL)

Input: Magnitude measurements $y_1, y_2, \ldots, y_m$.

1. Solve

$$\begin{array}{ll} \min\limits_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} & \left\|\boldsymbol{D}^T \boldsymbol{X} \boldsymbol{D}\right\|_{\ell_1} \\ \text{subject to} & y_i = \boldsymbol{a}_i^T \boldsymbol{X} \boldsymbol{a}_i,\ i=1,2,\ldots,m. \\ & \boldsymbol{X} \succeq 0. \end{array}$$

2. $\hat{\boldsymbol{x}} = \sigma_1 \boldsymbol{v}_1$, where σ_1 is the largest eigenvalue of the solution to the above optimization problem ($\hat{\boldsymbol{X}}$) and $\boldsymbol{v}_1$ is the corresponding eigenvector.

Output: Signal $\hat{\boldsymbol{x}}$ up to a global phase.

5 Efficient Implementation

In this section we present a method that reduces the computational complexity involved in Sparse Phase Lift. The main steps of the algorithm are depicted in 3. This derivation is inspired by the general scheme of [5, 1] adapted to a dual formulation for this problem. It should be noted that standard optimization methods can not be applied here. To be concrete, even when the size of the signal is as small as $n = 70$, the problem becomes prohibitive for standard Interior Point method solvers e.g. [4]. With the algorithm we present here we can handle signals of size $16,384$ (128×128 pixel images). Scaling this up to larger sizes is an important question that we hope to address in the future. In the next paragraphs we quickly go through the derivation of this algorithm. Since the objective of SPL is non-smooth we add a smoothing term of the form $\frac{1}{2}\mu\left\|\boldsymbol{X}-\boldsymbol{X}_0\right\|_F^2$. We also introduce the epigraph variable $t \in \mathbb{R}$.

Algorithm 3 Efficient Implementation of Sparse Phase Lift

Input: Magnitude measurements $y_1, y_2, \dots, y_m$, initial primal and dual solutions $\boldsymbol{X}_0, \boldsymbol{\Lambda}_0 \in \mathbb{R}^{n\times n}, \boldsymbol{\lambda}_0 \in \mathbb{R}^m, \boldsymbol{\Gamma}_0 \in \mathbb{R}^{N\times N}$, smoothing parameter μ, and step sizes $\{t_k\}$.

Initialization: $\theta_0 \leftarrow 1, \quad \tilde{\boldsymbol{\lambda}}_0 \leftarrow \boldsymbol{\lambda}_0, \quad \tilde{\boldsymbol{\Lambda}}_0 \leftarrow \boldsymbol{\Lambda}_0, \quad \tilde{\boldsymbol{\Gamma}}_0 \leftarrow \boldsymbol{\Gamma}_0.$

Iteration:

for $k = 0, 1, 2, \dots$ **do**

$\bar{\boldsymbol{\lambda}}_k \leftarrow (1-\theta_k)\tilde{\boldsymbol{\lambda}}_k + \theta_k \boldsymbol{\lambda}_k.$

$\bar{\boldsymbol{\Lambda}}_k \leftarrow (1-\theta_k)\tilde{\boldsymbol{\Lambda}}_k + \theta_k \boldsymbol{\Lambda}_k.$

$\bar{\boldsymbol{\Gamma}}_k \leftarrow (1-\theta_k)\tilde{\boldsymbol{\Gamma}}_k + \theta_k \boldsymbol{\Gamma}_k.$

$\boldsymbol{X}_k \leftarrow \boldsymbol{X}_0 + \frac{1}{\mu}\big(\bar{\boldsymbol{\Lambda}}_k + \boldsymbol{D}\bar{\boldsymbol{\Gamma}}_k\boldsymbol{D}^T - \sum_{i=1}^m (\bar{\lambda}_k)_i \boldsymbol{a}_i\boldsymbol{a}_i^T\big).$

$\boldsymbol{\lambda_{k+1}} \leftarrow \bar{\boldsymbol{\lambda}}_k - \frac{t_k}{\theta_k}\begin{bmatrix} y_1 - \boldsymbol{a}_1^T\boldsymbol{X}_k\boldsymbol{a}_1 \\ \vdots \\ y_m - \boldsymbol{a}_m^T\boldsymbol{X}_k\boldsymbol{a}_m \end{bmatrix}.$

$\boldsymbol{\Lambda}_{k+1} \leftarrow \mathcal{S}\big(\bar{\boldsymbol{\Lambda}}_k - \frac{t_k}{\theta_k}\boldsymbol{X}_k\big).$

$\boldsymbol{\Gamma}_{k+1} \leftarrow \mathcal{T}(\bar{\boldsymbol{\Gamma}}_k - \frac{t_k}{\theta_k}\boldsymbol{D}^T\boldsymbol{X}_k\boldsymbol{D}, \frac{t_k}{\theta_k})$

$\tilde{\boldsymbol{\lambda}}_{k+1} \leftarrow (1-\theta_k)\tilde{\boldsymbol{\lambda}}_k + \theta_k\boldsymbol{\lambda}_{k+1}.$

$\tilde{\boldsymbol{\Lambda}}_{k+1} \leftarrow (1-\theta_k)\tilde{\boldsymbol{\Lambda}}_k + \theta_k\boldsymbol{\Lambda}_{k+1}.$

$\tilde{\boldsymbol{\Gamma}}_{k+1} \leftarrow (1-\theta_k)\tilde{\boldsymbol{\Gamma}}_k + \theta_k\boldsymbol{\Gamma}_{k+1}.$

$\theta_{k+1} \leftarrow \frac{2}{(1+\sqrt{1+\frac{4}{\theta_k^2}})}.$

end for

Output: Signal $\hat{\boldsymbol{x}}$ up to a global phase.

Therefore, the optimization in SPL can be approximated by

$$\begin{aligned} \min_{t\in\mathbb{R},\ \boldsymbol{X}\in\mathbb{R}^{n\times n}} \quad & t + \tfrac{1}{2}\mu\|\boldsymbol{X} - \boldsymbol{X}_0\|_F^2 \\ \text{subject to} \quad & y_i = \boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{a}_i, \quad \|\boldsymbol{D}^T\boldsymbol{X}\boldsymbol{D}\|_{\ell_1} \le t. \end{aligned}$$

Putting this into the Lagrangian form we have

$$\begin{aligned} \max_{\nu\in\mathbb{R},\ \boldsymbol{\lambda}\in\mathbb{R}^m,\ \boldsymbol{\Lambda}\in\mathbb{R}^{n\times n},\ \boldsymbol{\Gamma}\in\mathbb{R}^{N\times N}} \ \min_{t\in\mathbb{R},\ \boldsymbol{X}\in\mathbb{R}^{n\times n}} \quad & t - \nu t + \frac{1}{2}\mu\|\boldsymbol{X} - \boldsymbol{X}_0\|_F^2 \\ & - \sum_{i=1}^m \lambda_i(y_i - \boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{a}_i) - \langle \boldsymbol{X}, \boldsymbol{\Lambda}\rangle - \langle \boldsymbol{D}^T\boldsymbol{X}\boldsymbol{D}, \boldsymbol{\Gamma}\rangle. \\ \text{subject to} \quad & \boldsymbol{\Lambda} \succeq \boldsymbol{0}, \quad \|\Gamma\|_{\ell_\infty} \le \nu. \end{aligned}$$

Notice that, the Lagrangian is unbounded unless $\nu = 1$, which results in

$$\begin{aligned} \max_{\boldsymbol{\lambda},\ \boldsymbol{\Lambda},\ \boldsymbol{\Gamma}} \ \min_{\boldsymbol{X}} \quad & \frac{1}{2}\mu\|\boldsymbol{X} - \boldsymbol{X}_0\|_F^2 - \sum_{i=1}^m \lambda_i y_i - \langle \boldsymbol{X}, \boldsymbol{\Lambda} + \boldsymbol{D}\boldsymbol{\Gamma}\boldsymbol{D}^T - \sum_{i=1}^m \lambda_i \boldsymbol{a}_i\boldsymbol{a}_i^T\rangle \\ \text{subject to} \quad & \boldsymbol{\Lambda} \succeq \boldsymbol{0}, \quad \|\Gamma\|_{\ell_\infty} \le 1. \end{aligned}$$

Define the dual function $g_\mu(\boldsymbol{\lambda}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma})$ as the objective of the above maximization problem. The minimizer $\boldsymbol{X}(\boldsymbol{\lambda}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma})$ is

$$\boldsymbol{X}(\boldsymbol{\lambda}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}) = \boldsymbol{X}_0 + \frac{1}{\mu}\big(\boldsymbol{\Lambda} + \boldsymbol{D}\boldsymbol{\Gamma}\boldsymbol{D}^T - \sum_{i=1}^m \lambda_i \boldsymbol{a}_i\boldsymbol{a}_i^T\big).$$

For maximizing the dual function we use a Nestrov style gradient update [5], i.e.

$$\begin{bmatrix} \boldsymbol{\lambda}_{k+1} \\ \boldsymbol{\Lambda}_{k+1} \\ \boldsymbol{\Gamma}_{k+1} \end{bmatrix} = \arg\min_{\boldsymbol{\lambda},\boldsymbol{\Lambda},\boldsymbol{\Gamma}} \left\| \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\Lambda} \\ \boldsymbol{\Gamma} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\lambda}_k \\ \boldsymbol{\Lambda}_k \\ \boldsymbol{\Gamma}_k \end{bmatrix} - \frac{t_k}{\theta_k}\nabla g_\mu(\boldsymbol{\lambda}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}) \right\|_{\ell_2}.$$

Notice that the updates are separable so we need to solve the following optimizations.

- **$\boldsymbol{\lambda}$ update:**

$$\begin{aligned} \boldsymbol{\lambda_{k+1}} &= \arg\min_{\boldsymbol{\lambda}} \ \frac{\theta_k}{2t_k}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_k\|_{\ell_2}^2 + \sum_{i=1}^m \lambda_i(y_i - \boldsymbol{a}_i^T\boldsymbol{X}_k\boldsymbol{a}_i), \\ \boldsymbol{\lambda_{k+1}} &= \boldsymbol{\lambda}_k - \frac{t_k}{\theta_k}\begin{bmatrix} y_1 - \boldsymbol{a}_1^T\boldsymbol{X}_k\boldsymbol{a}_1 \\ \vdots \\ y_m - \boldsymbol{a}_m^T\boldsymbol{X}_k\boldsymbol{a}_m \end{bmatrix}. \end{aligned}$$

- **$\boldsymbol{\Lambda}$ update:**

$$\begin{aligned} \boldsymbol{\Lambda}_{k+1} &= \arg\min_{\boldsymbol{\Lambda}:\ \boldsymbol{\Lambda}\succeq\boldsymbol{0}} \ \frac{\theta_k}{2t_k}\|\boldsymbol{\Lambda} - \boldsymbol{\Lambda}_k\|_F^2 + \langle \boldsymbol{\Lambda}, \boldsymbol{X}_k\rangle, \\ \boldsymbol{\Lambda}_{k+1} &= \mathcal{S}\big(\boldsymbol{\Lambda}_k - \frac{t_k}{\theta_k}\boldsymbol{X}_k\big), \end{aligned}$$

where $\mathcal{S}$ is the projection onto the positive semidefinite cone, which can be calculated by taking SVD and keeping the positive eigenvalues and their corresponding eigenvectors.

- **$\boldsymbol{\Gamma}$ update:**

$$\begin{aligned} \boldsymbol{\Gamma}_{k+1} &= \arg\min_{\boldsymbol{\Gamma}:\ \|\boldsymbol{\Gamma}\|_{\ell_\infty}\le 1} \ \frac{\theta_k}{2t_k}\|\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_k\|_F^2 + \langle \boldsymbol{\Gamma}, \boldsymbol{D}^T\boldsymbol{X}_k\boldsymbol{D}\rangle, \\ \boldsymbol{\Gamma}_{k+1} &= \mathcal{T}(\boldsymbol{\Gamma}_k - \frac{t_k}{\theta_k}\boldsymbol{D}^T\boldsymbol{X}_k\boldsymbol{D}, \frac{t_k}{\theta_k}), \end{aligned}$$

where the operator $\mathcal{T}(\boldsymbol{Z}, \tau)$ applies $\operatorname{sgn}(z)\cdot\min\{|z|, \tau\}$ to each element.

Notice that these are almost the update rules in Algorithm 3. We get the algorithm by adding Nestrov style moment updates for the dual variables and θ_k to make the algorithm converge faster. This is the reason for the auxilary variables with ˜ and ¯ on top.[2] We have used a backtracking method for the step sizes t_k.

6 Simulations on Synthetic Data

In this section we perform some simulations on synthetic data to better understand the regime of applicability of Phase Lift and Sparse Phase Lift.

6.1 Phase Lift vs. Sparse Phase Lift on a simple example

In this section we wish to demonstrate the performance of Sparse Phase Lift on a small synthetic image. For this purpose we generate a random $s = 5$-sparse signal of size 8×8,[3] and take its wavelet transform to get a synthetic image $\boldsymbol{x}$ which is spare in a wavelet frame. The wavelet frame we used was Coiflet from Stanford Wavelab [6]. We used $m = n$ random gaussian magnitude measurements. The synthetic image $\boldsymbol{x}$ with its sparse coefficients in the wavelet frame $\boldsymbol{s}$ are shown in Figure 5(a) and 5(b). Using Phase Lift and Sparse Phase Lift the reconstructed image $\hat{\boldsymbol{x}}$ with its sparse coefficients in the wavelet frame $\hat{\boldsymbol{s}}$ are shown in Figures 6(a), 6(b), 7(a) and 7(b). As can be seen Sparse Phase Lift is successful but Phase Lift fails. This is to be expected since Phase Lift does not utilize the low dimensional structure of the signal. In this example Phase Lift and Sparse Phase Lift achieved relative errors[4] of 0.8747 and 4.4523×10^{-5}.

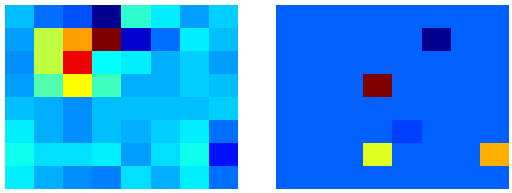
6.2 Phase Lift vs. Sparse Phase Lift in terms of minimum measurements

In this section we compare the number of measurements needed for Phase Lift and Sparse Phase Lift. For this purpose we consider an s-sparse signal $\boldsymbol{x} \in \mathbb{R}^n$. For now we assume $\boldsymbol{D} = \boldsymbol{I}$. The measurements $\boldsymbol{a}_i$ where chosen i.i.d. random vectors with $\mathcal{N}(0, 1)$ entries. The number of measurements m was varied from small to large. For each value of m, 100 trials were run. A value of m was considered successful if in at least 90 trials out of the total 100 trials that value of m resulted in successful recovery (this corresponds to probability of success ≥ 0.9). The smallest such value m for both methods

[2] Due to space limitations we we refer to [5, 1] for further details.

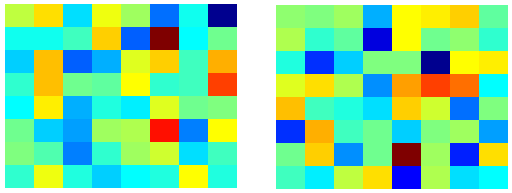
[3] By random sparse signal we mean that the support of the signal was chosen uniformly at random among the all possible $\binom{n}{s}$ possible supports of size s. The values on the support where chosen i.i.d. $\mathcal{N}(0, 1)$. Here, $n = 8 \times 8 = 64$.

[4] We calculate relative error using $\frac{\|\hat{\boldsymbol{X}} - \boldsymbol{X}\|_F}{\|\boldsymbol{X}\|_F}$.



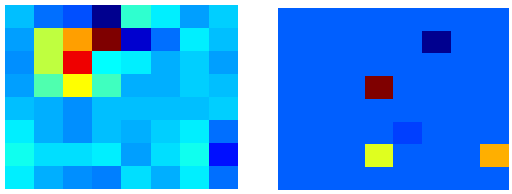
(a) Original. (b) Sparse coeff.

Figure 5



(a) Reconstructed image $\hat{\boldsymbol{x}}$. (b) Sparse coeff.

Figure 6: Phase Lift



(a) Reconstructed. (b) Sparse coeff.

Figure 7: Sparse Phase Lift

was tabulated in Table 1, for $n = 50$ and $s = 5, 10, 20$. As can be seen Sparse Phase Lift can achieve much lower sampling rates than PhaseLift by exploiting the sparsity of the signal. However, unlike compressed sensing where the number of measurements is roughly proportional to the sparsity level s, here the relationship seems to be quadratic ($\mathcal{O}(s^2)$).

s	Phase Lift	Sparse Phase Lift
2	122	18
5	122	26
10	122	80

Table 1: Comparison of minimum number of measurements m for different sparsity levels with $n = 50$.

7 Simulations on Real Images

As stated in previous sections, using our proposed efficient solver we are able to increase the range of applicability of Sparse Phase Lift. Despite this improvement, it takes a long time to perform Sparse Phase Lift on large image sizes. Therefore, due to time constraints, we chose 10 grayscale 512×512 images from the Miscellaneous category of the USC-SIPI Image Database [7] and down sampled them to 64×64.[5] We present in Table 2 the average relative errors of Phase Lift and Sparse Phase Lift for various forms of measurement mechanisms. The dictionary used form the images was Coiflet from [6]. The measurement methods are: **1-**$m = n$ random gaussian measurements. **2-**$m = n$ measurements consisting of columns of $\boldsymbol{F}$ where $\boldsymbol{F}$ is the DFT matrix. **3-** $m = 2n$ measurements consisting of columns of $\boldsymbol{F}$ and $\boldsymbol{FW}$, where $\boldsymbol{W}$ is a diagonal matrix with each diagonal entry a complex normal random variable.

[5]The reader should be reminded that this is equivalent to solving SDP of size 4096 which is time consuming even with our efficient solver.

Table 2: average relative error on 10 grayscale images from USC image data base (down-sampled to 64×64) using Coiflet wavelet of Stanford Wavelab.

Algorithm	Gaussian $m = n$	Fourier $m = n$
Phase Lift	0.94	0.97
Spare Phase Lift	0.09	0.92
Algorithm	$\boldsymbol{F}\&\boldsymbol{FW}$ $m = 2n$	$\boldsymbol{F}, \boldsymbol{FW}\&\boldsymbol{FW}'$ $m = 3n$
Phase Lift	0.73	0.008
Spare Phase Lift	0.06	0.003

4- $m = 3n$ measurements consisting of columns of $\boldsymbol{F}$ and $\boldsymbol{FW}$, and $\boldsymbol{FW}'$ where $\boldsymbol{W}$ and $\boldsymbol{W}'$ are independent diagonal matrices with each diagonal entry a complex normal random variable. As can be seen in the Table, for case (4), both algorithms work well. In case (1) and (3) Sparse Phase Lift shows superior performance. Again, this is to be expected since Phase Lift does not incorporate the sparsity of the images in wavelet domain. Both methods fail in (2). This says that Sparse Phase Lift is still not able to reconstruct the images from only magnitudes of the Fourier transform of the signal. However, Sparse Phase Lift still dramatically decreases the number of magnitude measurements needed.

Finally, we test Sparse Phase Lift on a boat image from the same data base. This time we down sample the image to size 128×128, and use case (3) for our measurements. The original image along with its reconstruction is shown in Figures 8(a) and 8(b) .

(a) Original. (b) Reconstructed.

Figure 8: Performance of Sparse Phase Lift on downsized boat with measurements $\boldsymbol{F}$ and $\boldsymbol{FW}$.

References

[1] S. Becker, E. J. Candès, and M. Grant. *Templates for convex cone problems with applications to sparse signal recovery.* To appear in, Mathematical Programming Computation 3(3), 165–218. 2010.

[2] A. Ben-Tal and A. Nemirovski *Lectures on Modern Convex Optimization, Analysis, Algorithms and Engineering Applications* Philadelphia, PA: SIAM, 2001.

[3] E. J. Candes, T. Strohmer, V. Voroninski *PhaseLift: Exact and Stable Signal Recovery from Magnitude Measurements via Convex Programming*, arXiv:1010.2955v2.

[4] M. Grant and S. Boyd *CVX: Matlab Software for Disciplined Convex Programming, version 1.21*, http://cvxr.com/cvx, Apr. 2011.

[5] Y. Nesterov. *Smooth minimization of non-smooth funcitons.* Math. Program., Serie A, 103:127-152, 2005.

[6] Available at http://www-stat.stanford.edu/~wavelab.

[7] USC-SIPI Image Database. Available at http://sipi.usc.edu/database/.