

PHASELESS SUBSPACE TRACKING

Seyedehsara Nayer and Namrata Vaswani

Iowa State University, Ames, IA, USA.

ABSTRACT

This work takes the first steps towards solving the “phaseless subspace tracking” (PST) problem. PST involves recovering a time sequence of signals (or images) from phaseless linear projections of each signal under the following structural assumption: the signal sequence is generated from a much lower dimensional subspace (than the signal dimension) and this subspace can change over time, albeit gradually. It can be simply understood as a dynamic (time-varying subspace) extension of the low-rank phase retrieval problem studied in recent work.

Index Terms— Phase retrieval, PCA, low-rank

1. INTRODUCTION

The Phase Retrieval (PR) problem occurs in many applications such as ptychography, crystallography, astronomy. The original PR problem involves recovering an n length signal \mathbf{x} from the magnitudes of its Discrete Fourier Transform (DFT) coefficients. Generalized PR (see [1] and [2]) replaces DFT by inner products with any set of measurement vectors, \mathbf{a}_i . Thus, the goal is to recover \mathbf{x} from $|\langle \mathbf{a}_i, \mathbf{x} \rangle|, i = 1, 2, \dots, m$. It is clear that, without extra assumptions, PR will require $m \geq n$. In recent works, structural assumptions such as sparsity (see [3, 4, 5]) or low-rank (see [2]) have been incorporated into the PR problem in order to reduce the number of measurements m required for exact or accurate recovery. Low-rank has been used in two ways. One is to assume that a single signal re-arranged as a matrix (or a single image) is itself approximately or exactly low-rank. The goal is to recover this low-rank “signal” from its phaseless linear projections [6, 7]. The second is to assume that a time sequence of signals (or vectorized images) together form a matrix that is well modeled as being low-rank. Each signal/image is one column of this matrix. The measurements are phaseless linear projections of each signal or image (each column of the matrix) [2]. This problem has been referred to as “Low-Rank Phase Retrieval (LRPR)” in [2] where it was first studied.

Another way to interpret the LRPR problem is as follows: a time sequence of signals $\mathbf{x}_t, t = 1, 2, \dots, d$, are generated from an unknown low dimensional subspace, i.e., $\mathbf{x}_t = \mathbf{U}\mathbf{b}_t$, where \mathbf{U} is an $n \times r$ basis matrix (tall matrix with mutually orthonormal columns) with $r \ll n$, and \mathbf{b}_t is an $r \times 1$ coefficients’ vector. For each $\mathbf{x}_t, t = 1, 2, \dots, d$, we have m phaseless measurements, $y_{i,t} = |\langle \mathbf{a}_{i,t}, \mathbf{x}_t \rangle|, i =$

1, 2, ... m , $t = 1, 2, \dots, d$. The goal is to either just recover the subspace, $\text{span}(\mathbf{U})$, or to recover both $\text{span}(\mathbf{U})$ and the coefficients and hence recover the signals \mathbf{x}_t ’s (equivalently, recover the low-rank matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d]$). The former problem can be called “phaseless PCA”, although the only known way to exactly recover $\text{span}(\mathbf{U})$ involves also recovering the \mathbf{b}_t ’s via iterative alternating minimization (or gradient descent); see the LRPR algorithms of [2].

Our problem: Dynamic LRPR or phaseless subspace tracking (PST). PST can be simply understood as the dynamic (time-varying subspace) extension of LRPR. Thus, instead of the subspace $\text{span}(\mathbf{U})$ being fixed, we assume that it can change with time, albeit slowly. Often, for long data sequences, e.g., long image sequences or videos, if one tries to use a single lower dimensional subspace to represent the entire data sequence, the required subspace dimension may end up being quite large. This can be problematic because it means that the resulting data matrix may not be sufficiently low-rank. In such cases, a better model is to assume that the data lies in a low-dimensional subspace that can change over time, albeit gradually.

The most general model for time-varying subspaces allows the subspace to change by a little at each time. However such a model involves too many unknowns. An r dimensional subspace in n -dimensional ambient space is fully specified by nr parameters. But the signal \mathbf{x}_t is an $n \times 1$ vector (has only n unknowns). Thus, allowing the subspace to change at each time will result in an increase in the number of unknowns (rather than a decrease which is the purpose of incorporating structure into the PR problem). A less general model, but one that allows for a reduction in the number of unknowns, is to assume that the true data subspace is piecewise constant with time. This model has been extensively used in robust subspace tracking literature [8, 9, ?] where it in fact helps ensure identifiability of the subspaces (in that problem, only one n length measurement vector is available at each time t).

Denote the subspace change times by t_j for $j = 1, 2, \dots, J$ and let $t_0 = 0$. Thus, we assume that $\mathbf{x}_t = \mathbf{U}_t \mathbf{b}_t$ where $\mathbf{U}_t = \mathbf{U}_{t_j}$ for all $t = t_j, t_j + 1, \dots, t_{j+1} - 1$. For simplicity, we sometimes misuse notation and use \mathbf{U}_j to denote \mathbf{U}_{t_j} . The goal is to recover the \mathbf{x}_t ’s from m phaseless measurements at each time, i.e., from $y_{i,t} := |\langle \mathbf{a}_{i,t}, \mathbf{x}_t \rangle|, i = 1, 2, \dots, m$ for each $t = 1, 2, \dots, d$. Under this model and assuming “slow subspace change” (quantified in Sec. 1.1), the question

is when can one solve this problem using a smaller m per signal than what is needed for LRPR? The LRPR work [2] has already demonstrated that just exploiting the low-rank assumption enables a reduction in the required m compared to simple PR done for each signal \mathbf{x}_t individually.

Our Contribution. This work takes the first steps towards solving the phaseless subspace tracking problem described above. Any subspace tracking problem requires two sub-problems to be solved:

- (a) given an accurate estimate of the previous subspace, detect if the subspace has changed; and
- (b) given an accurate estimate of the previous subspace, and given that it is known that the subspace *has* changed, estimate the new subspace using as few measurements (as short a tracking delay) as possible.

We will henceforth refer to (a) as “phaseless subspace change detection” and to (b) as “phaseless PCA with partial subspace knowledge”. Of course to solve (b) to ϵ accuracy for any $\epsilon \geq 0$, the resulting algorithm needs to also estimate the subspace projection coefficients as well. An early version of the current work (one that only solves (b) and only under the assumption that the previous subspace is exactly known) will be presented at Asilomar [10].

1.1. Notation, assumptions and some definitions

Notation. $\|\cdot\|$ denotes the l_2 norm of a vector or the induced l_2 norm of a matrix. For other l_p norms, we use $\|\cdot\|_p$.

A matrix with mutually orthonormal columns is referred to as a “*basis matrix*”. For basis matrices \hat{U}, U , the subspace error (SE) between their respective column spans is quantified by $\text{SE}(\hat{U}, U) := \|\left(I - \hat{U}\hat{U}'\right)U\|$. This measures the sine of the principal angle between the subspaces. The *phase-invariant distance* between two vectors is quantified using $\text{dist}(z_1, z_2) := \min_{\phi \in [0, 2\pi]} \|z_1 - e^{\sqrt{-1}\phi}z_2\|$. Normalized column-wise phase-invariant recovery error for matrices X and \hat{X} , its estimated version, is computed as $\text{NormErr}(X, \hat{X}) := \frac{\sum_{k=1}^q \text{dist}(\mathbf{x}_k, \hat{\mathbf{x}}_k)^2}{\sum_{k=1}^q \|\mathbf{x}_k\|^2}$. For any two integers i_1, i_2 , the interval $[i_1 : i_2]$ denotes the set of integer values $\{i_1, i_1 + 1, \dots, i_2\}$ and interval $[i_1 : i_2)$ denotes the set $\{i_1, i_1 + 1, \dots, i_2 - 1\}$.

Assumptions. We quantify “slow subspace change” using the model from [9]. In [9] and previous work, this has been successfully used to improve outlier tolerance of dynamic robust PCA as compared to its static counterpart. “Slow subspace change” [9] means that $U_t = U_{t_j} := U_j \forall t \in [t_j : t_{j+1})$ (piecewise constant subspaces) and the following hold:

- (a) $\text{SE}(U_{j-1}, U_j) \leq \Delta$ with Δ small,
- (b) at each change time, only one direction changes, and
- (c) $\min_j(t_{j+1} - t_j)$ is lower bounded.

Definitions. As mentioned above, at each subspace change time, only one direction changes, while the rest of the subspace remains fixed. Of course at different change times, the

changing direction could be different, thus over a long period of time, the entire subspace could change. To explain this further, if $u_{j-1,\text{chg}}$ denotes the direction from $\text{span}(U_{j-1})$ that changes at t_j , and $u_{j,\text{chd}}$ denotes its changed version, then $\text{span}(U_{j-1}) = \text{span}([U_{j-1,\text{fix}}, u_{j-1,\text{chg}}])$ and $\text{span}(U_j) = \text{span}([U_{j-1,\text{fix}}, u_{j,\text{chd}}])$, where $U_{j-1,\text{fix}}$ is an $n \times (r-1)$ matrix corresponding to the fixed part of the subspace at t_j . Denote the direction that gets added to the subspace at time t_j by $u_{j,\text{add}}$. Clearly,

$$u_{j,\text{add}} := \frac{(I - u_{j-1,\text{chg}}u_{j-1,\text{chg}}')u_{j,\text{chd}}}{\text{SE}(u_{j-1,\text{chg}}, u_{j,\text{chd}})}.$$

Also, $\theta_j := \cos^{-1}|u_{j-1,\text{chg}}'u_{j,\text{chd}}|$ is the angle between $u_{j-1,\text{chg}}$ and $u_{j,\text{chd}}$ and $u_{j,\text{del}} := u_{j-1,\text{chg}} \sin \theta_j - u_{j,\text{add}} \cos \theta_j$ is the direction that gets deleted at t_j .

The following facts are immediate from the above: (i) $|\sin \theta_j| = \sin \theta_j = \text{SE}(u_{j-1,\text{chg}}, u_{j,\text{chd}})$, (ii) $u_{j,\text{chd}} = u_{j,\text{add}} \sin \theta_j + u_{j-1,\text{chg}} \cos \theta_j$, (iii) $u_{j,\text{add}}$ is orthogonal to U_{j-1} , and (iv) $\text{span}(U_j) \subseteq \text{span}([U_{j-1}, u_{j,\text{add}}])$.

Define the sub-matrix $X_j := [\mathbf{x}_{t_j}, \mathbf{x}_{t_j+1}, \dots, \mathbf{x}_{t_{j+1}-1}]$, let $q := (t_{j+1} - t_j)$ and let

$$\mathbb{E} \left[\frac{1}{q} \sum_{t \in [t_j, t_{j+1}]} \mathbf{x}_t \mathbf{x}_t' \right] \stackrel{\text{EVD}}{=} U_j \bar{\Lambda}_j U_j'$$

denote its eigenvalue decomposition (EVD). To simplify notation, in the text below we sometimes remove the subscript j , e.g., we often use θ to denote θ_j .

2. SOLUTION APPROACH

In the next two subsections, we explain how to solve each of the two sub-problems mentioned above.

2.1. Automatic phaseless subspace change detection

Consider the matrices

$$Y_U := \frac{1}{mq} \sum_{i=1}^m \sum_{t \in [t_j : t_{j+1}]} y_{i,t}^2 \mathbf{a}_{i,t} \mathbf{a}_{i,t}', \text{ and} \quad (1)$$

$$\tilde{Y}_U := (I - \hat{U}_{j-1} \hat{U}_{j-1}') Y_U (I - \hat{U}_{j-1} \hat{U}_{j-1}'). \quad (2)$$

To understand our approach simply, suppose that \hat{U}_{j-1} is a perfect estimate, i.e., suppose that $\text{span}(\hat{U}_{j-1}) = \text{span}(U_{j-1})$. Then, it is not hard to see that

$$\begin{aligned} \mathbb{E}[\tilde{Y}_U] &= (I - U_{j-1} U_{j-1}') [2(U_j \bar{\Lambda}_j U_j') + \text{tr}(\bar{\Lambda})I] (I - U_{j-1} U_{j-1}') \\ &= (2 \sin^2 \theta \lambda_{\min}(\bar{\Lambda})) u_{j,\text{add}} u_{j,\text{add}}' + \text{tr}(\bar{\Lambda})(I - U_{j-1} U_{j-1}'). \end{aligned} \quad (3)$$

The first equality follows from [11, Lemma A.1], and the second follows using the subspace change assumption. Observe that the above matrix is orthogonal to U_{j-1} . Let $U_{j-1,\perp}$ be a basis matrix for the subspace orthogonal to U_{j-1} . Since U_{j-1} has rank r , this will be an $n \times (n-r)$ matrix. Since $u_{j,\text{add}}$ is orthogonal to U_{j-1} , thus, without loss of generality, we can assume that $u_{j,\text{add}}$ is one of the columns of $U_{j-1,\perp}$.

Denote the matrix for the rest of its columns by $\tilde{\mathbf{U}}_{j-1,\perp}$. Thus, $\text{span}(\mathbf{U}_{j-1,\perp}) = \text{span}([\mathbf{u}_{j,\text{add}}, \tilde{\mathbf{U}}_{j-1,\perp}])$ and so, using (3), an EVD of $\mathbb{E}[\tilde{\mathbf{Y}}_U]$ is $\mathbb{E}[\tilde{\mathbf{Y}}_U] \stackrel{\text{EVD}}{=}$

$$[\mathbf{u}_{j,\text{add}}, \tilde{\mathbf{U}}_{j-1,\perp}] \begin{bmatrix} 2\sin^2\theta\lambda_{\min}(\bar{\Lambda}) + \text{tr}(\bar{\Lambda}) & 0 \\ 0 & \text{tr}(\bar{\Lambda})I \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j,\text{add}}' \\ \tilde{\mathbf{U}}_{j-1,\perp}' \end{bmatrix}.$$

Clearly the top eigenvector of this matrix is equal to $\mathbf{u}_{j,\text{add}}$, with the corresponding eigenvalue of $2\sin^2\theta\lambda_{\min}(\bar{\Lambda}) + \text{tr}(\bar{\Lambda})$, and a gap of $2\sin^2\theta\lambda_{\min}(\bar{\Lambda})$ between first and other eigenvalues. So, by law of large numbers [12], with high probability (w.h.p.), that the top eigenvector of this matrix will be a good initial estimation of $\mathbf{u}_{j,\text{add}}$, when m and q are large enough. Currently we are making an intuitive argument, these statements will be made rigorous in follow-up work.

From above, again by law of large numbers and assuming $\hat{\mathbf{U}}_{j-1}$ is a good estimate of \mathbf{U}_{j-1} , when m and q are large, if the subspace has not changed, the first eigenvalue of $\tilde{\mathbf{Y}}_U$, $\lambda_1(\tilde{\mathbf{Y}}_U)$, will be close to $\text{tr}(\bar{\Lambda})$ w.h.p.; while if it has changed, it will be close to $2\sin^2\theta\lambda_{\min}(\bar{\Lambda}) + \text{tr}(\bar{\Lambda})$ w.h.p.. A natural subspace change detection approach thus involves thresholding $\lambda_1(\tilde{\mathbf{Y}}_U)$. Thus, $\lambda_1(\tilde{\mathbf{Y}}_U) \geq C\text{tr}(\bar{\Lambda})$ can be used as a criterion for detecting the change with C being a value slightly more than one. Here $\text{tr}(\bar{\Lambda})$ is unknown but notice that for $i = r+1, r+2, \dots, n$, $\lambda_i(\mathbb{E}[\mathbf{Y}_U]) = \text{tr}(\bar{\Lambda})$. Thus, w.h.p., when m and q are large enough, $\lambda_n(\mathbf{Y}_U) \approx \text{tr}(\bar{\Lambda})$ and so we use $\lambda_n(\mathbf{Y}_U)$ as an estimate of $\text{tr}(\bar{\Lambda})$. Algorithm 1 summarizes our approach.

It is clear from the above that the change detection performance improves as θ increases. This fact is also observed through our experiments (see the ROC curves in Fig. 1).

2.2. Phaseless PCA with partial subspace knowledge

After detecting the change, the next step of PST algorithm is estimating the current subspace knowing the existence of a change. Here in order to use the previously estimated subspace $\hat{\mathbf{U}}_{j-1}$, we construct a bigger subspace matrix $\tilde{\mathbf{U}}_j \in \mathbb{R}^{n \times (r+1)}$ which contains a new added column besides $\hat{\mathbf{U}}_{j-1}$. Similarly, the number of rows of \mathbf{B}_j are increased by one and $\tilde{\mathbf{B}}_j \in \mathbb{R}^{(r+1) \times q}$ is a relaxed estimation of \mathbf{B}_j . Phaseless PCA with partial subspace knowledge consists of two steps which we explain in the following.

Initialization. The initialization is inspired by the previously proposed spectral method which is used in many existing works like [1] and extended in [2]. From the discussion above, when m and q are large enough and $\text{span}(\hat{\mathbf{U}}_{j-1})$ is close to $\text{span}(\mathbf{U}_{j-1})$, using the Davis-Kahan $\sin\theta$ theorem [13], it can be argued that the top eigenvector of $\tilde{\mathbf{Y}}_U$ will be a good estimate of $\mathbf{u}_{j,\text{add}}$. Denote this by $\hat{\mathbf{u}}_{j,\text{add}}$. With this, $\tilde{\mathbf{U}}_j = [\hat{\mathbf{U}}_{j-1} \ \hat{\mathbf{u}}_{j,\text{add}}]$ can be used as the initial estimated subspace. Using an idea similar to the approach of [2], the top eigenvector of

$$\mathbf{Y}_b = \tilde{\mathbf{U}}_j' \left(\frac{1}{m} \sum_{i=1}^m y_{i,t}^2 \mathbf{a}_{i,t} \mathbf{a}_{i,t}' \right) \tilde{\mathbf{U}}_j, \quad (4)$$

Algorithm 1 PST-detection

- 1: Compute \mathbf{Y}_U and $\tilde{\mathbf{Y}}_U$ using (1) and (2).
- 2: Declare a change if $\lambda_1(\tilde{\mathbf{Y}}_U) \geq C \lambda_n(\mathbf{Y}_U)$.

denoted $\hat{\mathbf{b}}_t$, will be a good estimate of $\tilde{\mathbf{b}}_t := \begin{bmatrix} \mathbf{b}_{t,1:r-1} \\ b_{t,r} \cos \theta \\ b_{t,r} \sin \theta \end{bmatrix}$.

Main loop. Main loop is an alternating minimization solution. Similar to our previous work [2], this part has three steps. At each step one of the three variables is estimated and the other two is assumed to be constant. Using alternating minimization, $\tilde{\mathbf{U}}$ can be obtained by solving

$$\tilde{\mathbf{U}} = \arg \min_{\mathbf{U}} \sum_t \|\hat{\mathbf{C}}_t \mathbf{y}_t - \mathbf{A}_t' \mathbf{U} \hat{\mathbf{b}}_t\|^2,$$

where $\hat{\mathbf{b}}_t \in \mathbb{R}^{r+1}$ is provided by the previous iteration. In the simple situation where just one direction is changing at a time, recovering $\hat{\mathbf{u}}_{j,\text{add}}$ is enough. This can be obtained by

$$\hat{\mathbf{u}}_{j,\text{add}} = \arg \min_{\hat{\mathbf{u}}} \sum_t \|\hat{\mathbf{C}}_t \mathbf{y}_t - \mathbf{A}_t' \hat{\mathbf{U}}_0 \hat{\mathbf{b}}_{t,1:r} - \mathbf{A}_t' \hat{\mathbf{u}} \hat{\mathbf{b}}_{t,r+1}\|^2,$$

where $\hat{\mathbf{b}}_{t,1:r}$ and $\hat{\mathbf{b}}_{t,r+1}$ are vectors that contain the first r elements and the last element of $\hat{\mathbf{b}}_t$ respectively. Then matrix $\tilde{\mathbf{U}}_j = [\hat{\mathbf{U}}_{j-1} \ \hat{\mathbf{u}}_{j,\text{add}}]$ will be the estimate of current subspace. Rest of the solution is similar to our previous work. Using the new estimate of $\tilde{\mathbf{U}}_j$, $\tilde{\mathbf{B}}_j$ and the phase are updated respectively. To remove the relaxed dimension whenever is needed, singular value decomposition can be used as the last step. Matrix $\hat{\mathbf{X}}_j$ can be recovered as a byproduct also.

The complete algorithm is summarized in Algorithm 2. This works well when the column span of $\hat{\mathbf{U}}_{j-1}$ is a good estimate of the subspace spanned by \mathbf{U}_{j-1} . Its final subspace recovery error is lower bounded by $\text{SE}(\tilde{\mathbf{U}}_{j-1}, \mathbf{U}_{j-1})$. To reduce the error beyond this value, at the end of Algorithm 2, a few iterations of LRPR-AltMin (the algorithm of [2]) can be used. Just a few iterations of LRPR-AltMin will significantly reduce the error because this can be interpreted as beginning LRPR-AltMin with a very good initial estimate.

3. NUMERICAL EXPERIMENTS

Phaseless subspace change detection. To evaluate the detection performance of Algorithm 1, we plot the Receiver Operating Characteristic (ROC) curve by varying the constant C . 50 runs of two sets of data are generated, one without the change and one with the change. For each value of C , we compute the Monte Carlo estimate of the probability of correct detection by using the dataset with change, and we estimate the false alarm probability by using the dataset without change. We then plot the detection probability on the y-axis and the false alarm on the x-axis for various values of C . We show the plots in Fig. 1 for various values of θ . Settings for

Algorithm 2 PST-PCA

Input: $y_{i,t}$ and $\mathbf{a}_{i,t}$ for $i \in [1 : m], t \in [t_j : t_{j+1}]$, $\hat{\mathbf{U}}_{j-1}, r, n$

- 1: **Initialization:**
- 2: Compute $\tilde{\mathbf{Y}}_U$ as described in (2) and compute $\hat{\mathbf{u}}_{\text{add}}$ as its top eigenvector.
- 3: Set $\tilde{\mathbf{U}} = [\hat{\mathbf{U}}_{j-1}, \hat{\mathbf{u}}_{\text{add}}]$.
- 4: **for** $t \in [t_j, t_{j+1}]$ **do**
- 5: Compute \mathbf{Y}_b as described in (4).
- 6: Set $\hat{\tilde{\mathbf{b}}}_t$ as top eigenvector of \mathbf{Y}_b and scale it by $\sqrt{\frac{\sum_i y_{i,t}^2}{m}}$.
- 7: **end for**
- 8: **for** $\tau \in [0 : T_{\max})$ **do**
- 9: **for** $t \in [t_j, t_{j+1}]$ **do**
- 10: set $\hat{\mathbf{C}}_t = \text{phase}(\mathbf{A}_t' \tilde{\mathbf{U}} \hat{\tilde{\mathbf{b}}}_t)$.
- 11: Set $\hat{\tilde{\mathbf{b}}} = \begin{bmatrix} \hat{\tilde{\mathbf{b}}}_{[1:r],t} \\ \hat{\tilde{\mathbf{b}}}_{r+1,t} \end{bmatrix}$.
- 12: Set $\mathbf{d}_t = \hat{\mathbf{C}}_t \mathbf{y}_t - \mathbf{A}_t' \hat{\mathbf{U}}_{j-1} \hat{\tilde{\mathbf{b}}}_{[1:r],t}$
- 13: **end for**
- 14: Compute $\hat{\mathbf{u}}_{\text{add}}$ as
$$\left(\sum_{t \in [t_j : t_{j+1}]} (\hat{\tilde{\mathbf{b}}}_{r+1,t})^2 \mathbf{A}_t \mathbf{A}_t' \right)^{-1} \sum_{t \in [t_j : t_{j+1}]} \hat{\tilde{\mathbf{b}}}_{r+1,t} \mathbf{A}_t \mathbf{d}_t.$$
- 15: Set $\tilde{\mathbf{U}} = [\hat{\mathbf{U}}_{j-1}, \hat{\mathbf{u}}_{\text{add}}]$.
- 16: **for** $t \in [t_j, t_{j+1}]$ **do**
- 17: set $\hat{\tilde{\mathbf{b}}}_t = \text{argmin}_{\mathbf{b}} \|\hat{\mathbf{C}}_t \mathbf{y}_t - \mathbf{A}_t' \tilde{\mathbf{U}} \mathbf{b}\|$
- 18: **end for**
- 19: **end for**
- 20: Compute $\tilde{\mathbf{X}}_j = \tilde{\mathbf{U}} \hat{\tilde{\mathbf{b}}}$.

Output: Set $\hat{\mathbf{U}}_j$ as top r eigenvectors of $\tilde{\mathbf{X}}_j$, and $\hat{\mathbf{B}}_j = \hat{\mathbf{U}}_j' \tilde{\mathbf{X}}_j$.

this experiment are $n = 1000$, $r = 10$, $m = 850$, $q = 750$, $\theta = 30, 45, 60, 75$ degrees and values of C varying between 0 and 3, with intervals of 10^{-4} . Also $\hat{\mathbf{U}}_0$ was generated so that $\text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0) \approx 10^{-4}$.

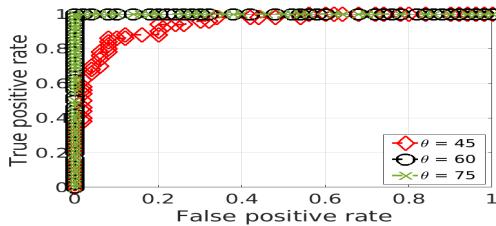


Fig. 1: ROC curve when $\text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0) \approx 10^{-4}$. “True positives” refers to probability of correct detection; “False positives” refers to false alarm probability.

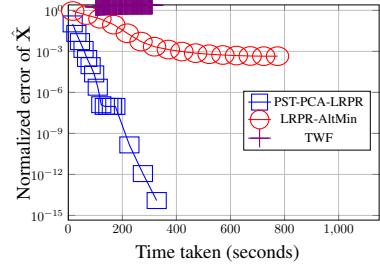


Fig. 2: Normalized recovery error of \mathbf{X} , $\text{NormErr}(\mathbf{X}, \hat{\mathbf{X}})$.

	$\text{SE} \approx 10^{-4}$			$\text{SE} \approx 10^{-6}$			
	m	$q=400$	$q=500$	$q=600$	$q=400$	$q=500$	$q=600$
450		0.84	0.92	0.98	0.92	0.98	1.00
550		1.00	1.00	1.00	1.00	1.00	1.00
650		1.00	1.00	1.00	0.98	1.00	1.00

Table 1: Success probabilities for two values of $\text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0)$. Cases with error of \mathbf{U} less than $1.5 \times \text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0)$ are considered successful. PST loop is broken when the subspace error of recovered \mathbf{U} is less than $1.5 \times \text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0)$, or when the difference of estimated values of \mathbf{U} between two successive iterations is less than 10^{-9} .

Phaseless PCA with partial subspace knowledge. In our first experiment, we let $n = 1000$, $r = 10$, $\theta = 30$ degrees. For two values of the initial subspace error $\text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0)$, and many values of q and m , we implemented PST-PCA and computed the probability of the final recovery error reaching the same level as the initial subspace error. This is displayed in Table 1. 50 Monte Carlo runs were used. In our second experiment, we compare PST-PCA performance with that of LRPR-AltMin (algorithm of [2]) and with Truncated Wirtinger Flow (TWF) [1]. TWF is one of the best known single signal PR algorithms provably requiring only $O(n)$ measurements for exact recovery. In this experiment, we implemented PST-PCA for 12 iterations followed by LRPR-AltMin for 3 iterations. LRPR-AltMin was implemented for 15 total iterations. We show our results in Fig. 2 where we plot the normalized phase-invariant recovery error of \mathbf{X} against the time taken in seconds. This is done by computing the recovery error and time taken at the end of each algorithm iteration. As can be seen just PST-PCA already significantly outperforms both LRPR-AltMin and TWF (just PST-PCA error decreases to 10^{-7} while that of LRPR-AltMin saturates at 10^{-3} and TWF is even worse). This experiment used $n = 1000$, $q = 500$, $r = 15$, $m = 700$, $\theta = 30$, and 50 Monte-Carlo repeats; and value of \mathbf{U}_0 is corrupted by an additive Gaussian noise so that $\text{SE}(\hat{\mathbf{U}}_0, \mathbf{U}_0) \approx 3.5 \times 10^{-4}$.

4. REFERENCES

- [1] Y. Chen and E. Candes, “Solving random quadratic systems of equations is nearly as easy as solving linear systems,” in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2015, pp. 739–747.
- [2] N. Vaswani, S. Nayer, and Y. C. Eldar, “Low rank phase retrieval,” *IEEE Trans. Sig. Proc.*, August 2017.
- [3] K. Jaganathan, S. Oymak, and B. Hassibi, “Recovery of sparse 1-d signals from the magnitudes of their fourier transform,” in *IEEE Intl. Symp. on Information Theory (ISIT)*. IEEE, 2012, pp. 1473–1477.
- [4] Y. Shechtman, A. Beck, and Y. C. Eldar, “Gespar: Efficient phase retrieval of sparse signals,” *IEEE Trans. Sig. Proc.*, vol. 62, no. 4, pp. 928–938, 2014.
- [5] A. Szameit, Y. Shechtman, E. Osherovich, E. Bullkitch, P. Sidorenko, H. Dana, S. Steiner, E. B. Kley, S. Gazit, T. Cohen-Hyams, S. Shoham, M. Zibulevsky, I. Yavneh, Y. C. Eldar, O. Cohen, and M. Segev, “Sparsity-based single-shot subwavelength coherent diffractive imaging,” *Nature Materials*, vol. 11, pp. 455–9, Apr. 2012.
- [6] S. Tu, R. Boczar, M. Soltanolkotabi, and B. Recht, “Low-rank solutions of linear matrix equations via procrustes flow,” *arXiv preprint arXiv:1507.03566*, 2015.
- [7] M.D. Butala, R.A. Frazin, Y. Chen, and F. Kamalabadi, “A monte carlo technique for large-scale dynamic tomography,” in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2007.
- [8] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, “Recursive robust pca or recursive sparse recovery in large but structured noise,” *IEEE Trans. Info. Th.*, pp. 5007–5039, August 2014.
- [9] P. Narayananamurthy and N. Vaswani, “Provable dynamic robust pca or robust subspace tracking,” *IEEE Trans. Info. Theory*, to appear, 2018.
- [10] S. Nayer and N. Vaswani, “Phaseless subspace tracking: Low-rank phase retrieval with partial subspace knowledge,” in *Signals, Systems, and Computers, 2018 52st Asilomar Conference*, 2018.
- [11] E. J. Candes, X. Li, and M. Soltanolkotabi, “Phase retrieval via wirtinger flow: Theory and algorithms,” *IEEE Trans. Info. Th.*, vol. 61, no. 4, pp. 1985–2007, 2015.
- [12] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *Compressed sensing*, pp. 210–268, 2012.
- [13] C. Davis and W. M. Kahan, “The rotation of eigenvectors by a perturbation. iii,” *SIAM J. Numer. Anal.*, vol. 7, pp. 1–46, Mar. 1970.