Low Rank Phase Retrieval

Namrata Vaswani (Iowa State Univ), Seyedehsara Nayer (Iowa State Univ), Yonina C. Eldar (Technion)

Problem Setting. In recent years there has been a large amount of work on the generalized phase retrieval (PR) problem: recover an n-length discrete signal (vector) $\boldsymbol{x}$ from measurements $|\boldsymbol{a}_i{}'\boldsymbol{x}|^2$, $i = 1, 2, \ldots, m$ [1], [2], [3], [4], [5]. Of these, a recent iterative algorithm, truncated Wirtinger Flow (TWF) [5], achieves the best computational and sample complexity. It can recover x from only $m \geq cn$ measurements. Here, and throughout, the letter c is reused to denote *different* numerical constants. Two recent modifications of TWF [6], [7] have the same order complexities but better empirical performance. By imposing structure on $\boldsymbol{x}$, the sample complexity can be improved. Most work on this direction imposes sparsity, e.g., [8].

In this abstract and [10], we study *"low-rank phase retrieval (LRPR)"* defined as follows. Instead of a single vector $\boldsymbol{x}$, we have a set of q vectors, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q$ that are such that the $n \times q$ matrix,

$$\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q],$$

has rank $r \ll \min(n, q)$. For each $\boldsymbol{x}_k$, we observe a set of m measurements of the form

$$\boldsymbol{y}_{i,k} := |\boldsymbol{a}_{i,k}{}'\boldsymbol{x}_k|^2,\ i = 1, 2, \ldots m,\ k = 1, 2, \ldots, q. \tag{1}$$

The measurement vectors, $\boldsymbol{a}_{i,k}$, are mutually independent. The goal is to recover the matrix $\boldsymbol{X}$ from these mq measurements $\boldsymbol{y}_{i,k}$. In some applications, the goal may be to only recover the span of the columns of $\boldsymbol{X}$, $\mathrm{range}(\boldsymbol{U})$. We call this the *phaseless PCA* problem.

Contribution 1: Algorithms. We develop two iterative algorithms for solving the LRPR problem. Our solution approach relies on the fact that a rank r matrix $\boldsymbol{X}$ can always be expressed (non-uniquely) as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$ where $\boldsymbol{U}$ is an $n \times r$ matrix with mutually orthonormal columns. The first step of both methods is a spectral initialization step, motivated by TWF [5], for initializing $\boldsymbol{U}$, and then, the columns of $\boldsymbol{B}$. We summarize this in Algorithm 1 (LRPR-init). The notation $\mathbb{1}_{\mathcal{E}}$ refers to the indicator function on the event $\mathcal{E}$. LRPR-init relies

Algorithm 1 Low Rank PR Initialization (LRPR-init)

Set $\hat{r} = \arg\max_j(\lambda_j(\boldsymbol{Y}_U) - \lambda_{j+1}(\boldsymbol{Y}_U))$ with

$$\boldsymbol{Y}_U := \frac{1}{mq}\sum_{i=1}^{m}\sum_{k=1}^{q}\boldsymbol{y}_{i,k}\boldsymbol{a}_{i,k}\boldsymbol{a}_{i,k}{}'\mathbb{1}_{\left\{y_{i,k} \leq 9\frac{\sum_i y_{i,k}}{\tilde{m}}\right\}}.$$

1) Compute $\hat{\boldsymbol{U}}$ as the top $\hat{r}$ eigenvectors of $\boldsymbol{Y}_U$.
2) For each $k = 1, 2, \ldots, q$,
   a) compute $\hat{\boldsymbol{v}}_k$ as the top eigenvector of $\boldsymbol{Y}_{b,k} := \hat{\boldsymbol{U}}'\boldsymbol{M}_k\hat{\boldsymbol{U}}$ where $\boldsymbol{M}_k := \frac{1}{m}\sum_i \boldsymbol{y}_{i,k}\boldsymbol{a}_{i,k}\boldsymbol{a}_{i,k}{}'$.
   b) compute $\hat{\nu}_k := \sqrt{\frac{1}{m}\sum_i \boldsymbol{y}_{i,k}}$; set $\hat{\boldsymbol{b}}_k = \hat{\boldsymbol{v}}_k\hat{\nu}_k$

Output $\hat{\boldsymbol{U}}$. Output $\hat{\boldsymbol{x}}_k := \hat{\boldsymbol{U}}\hat{\boldsymbol{b}}_k$ for all $k = 1, 2, \ldots, q$.

on two key ideas. First, $\mathbb{E}[\boldsymbol{Y_U}] = \frac{c_1}{q}\boldsymbol{X}\boldsymbol{X}' + c_2\boldsymbol{I}$, where c_1 and c_2 are positive scalars, and thus, the span of its top r eigenvectors is equal to $\mathrm{range}(\boldsymbol{U})$. Hence, by law of large numbers [11], with mq large enough, we expect that $\mathrm{range}(\hat{\boldsymbol{U}}) \approx \mathrm{range}(\boldsymbol{U})$. Second, if $\hat{\boldsymbol{U}}$ is independent of $\boldsymbol{M}_k$, then $\mathbb{E}[\boldsymbol{Y}_{b,k}|\hat{\boldsymbol{U}}] = \hat{\boldsymbol{U}}'[c_3\boldsymbol{x}_k\boldsymbol{x}_k' + c_4\boldsymbol{I}]\hat{\boldsymbol{U}}$. If $\hat{\boldsymbol{U}} = \boldsymbol{U}$, then $\hat{\boldsymbol{U}}'\boldsymbol{x}_k = \boldsymbol{b}_k$ and, in this case, the top eigenvector of $\mathbb{E}[\boldsymbol{Y}_{b,k}|\hat{\boldsymbol{U}}]$ is proportional to $\boldsymbol{b}_k$. With just $\mathrm{range}(\hat{\boldsymbol{U}}) \approx \mathrm{range}(\boldsymbol{U})$, this may not hold, but $\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}'\boldsymbol{x}_k$ will still be a good approximation of $\boldsymbol{x}_k$. By law of large numbers [11], the same will be true for $\hat{\boldsymbol{x}}_k$ given in LRPR-init. We show the power of LRPR-init in Fig. 1a.

The remainder of the algorithm is developed in one of two ways: using a projected gradient descent strategy to modify the TWF iterates (LRPR1); or developing an alternating minimization algorithm, motivated by AltMinPhase [3], that directly exploits the decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$ (LRPR2). Via extensive experiments, we show that both LRPR1 and LRPR2 have better sample complexity than TWF; with LRPR2 exhibiting the best performance; see Figs. 1b, 1c. Both also significantly outperform TWFproj (project TWF initialization onto space of rank r matrices, and do the same for each TWF iteration). We show their power for recovering an approximately low rank video sequence from CDP measurements in Figs. 2, 3.

Contribution 2: Initialization Error Guarantees. Our most important contribution is sample complexity bounds for LRPR-init. We state our key result next. It assumes that we use different (independent) sets of measurements for recovering $\boldsymbol{U}$ and $\boldsymbol{B}$. This ensures independence between $\hat{\boldsymbol{U}}$ and $\boldsymbol{M}_k$ in Algorithm 1.

Define $\rho := \frac{\max_k \|\boldsymbol{x}_k\|_2^2}{\frac{1}{q}\sum_k \|\boldsymbol{x}_k\|_2^2}$, let $\frac{1}{q}\boldsymbol{X}\boldsymbol{X}' \overset{\mathrm{EVD}}{=} \boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}'$ be its reduced eigenvalue decomposition (EVD), and let κ be its condition number. Also, let $\mathcal{N}(0, \boldsymbol{\Sigma})$ denote a Gaussian distribution with zero mean and covariance $\boldsymbol{\Sigma}$ and let $\mathrm{dist}(\boldsymbol{z}_1, \boldsymbol{z}_2) := \min_{\phi\in[0,2\pi]}\|\boldsymbol{z}_1 - e^{\sqrt{-1}\phi}\boldsymbol{z}_2\|_2$ denote the phase-invariant distance between two vectors.

Theorem 1. *For each column k of the rank r matrix $\boldsymbol{X}$, we observe, $\boldsymbol{y}_{i,k} := (\boldsymbol{a}_{i,k}{}'\boldsymbol{x}_k)^2$ where $\boldsymbol{a}_{i,k} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, \boldsymbol{I})$, for $i = 1, 2, \ldots, m$. For each k, we also observe $\boldsymbol{y}_{i,k}^B := (\boldsymbol{a}_i^B{}'\boldsymbol{x}_k)^2$ where $\boldsymbol{a}_i^B \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, \boldsymbol{I})$, for $i = 1, 2, \ldots, \tilde{m}$. The sets of vectors $\{\boldsymbol{a}_i^B\}_{i=1,2,\ldots,\tilde{m}}$ and $\{\boldsymbol{a}_{i,k}\}_{i=1,2,\ldots,m,\ k=1,2,\ldots,q}$ are mutually independent. Consider Algorithm 1 with $\boldsymbol{y}_{i,k}$ replaced by $\boldsymbol{y}_{i,k}^B$ in step 2.*

Assume that $\bar{\boldsymbol{\Lambda}}$ is such that $\bar{\lambda}_j - \bar{\lambda}_{j+1} \leq 0.9\bar{\lambda}_{\min}$ and $\kappa \leq 10$. Assume that $r \leq cn^{1/5}$. For an $\varepsilon < 1$, if

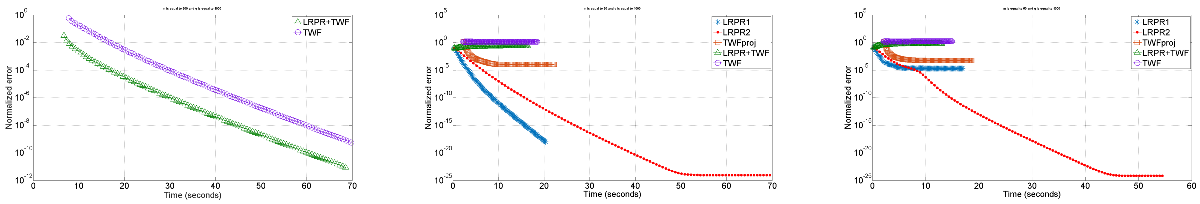
$$\tilde{m} \geq \frac{c\sqrt{n}}{\varepsilon^2},\ m \geq \frac{c\kappa^2 \cdot r^4 \log n (\log\tilde{m})^2}{\varepsilon^2},\ mq \geq \frac{c\rho^2\kappa^2 \cdot nr^4(\log\tilde{m})^2}{\varepsilon^2},$$

then, with probability at least $1 - 2\exp(-cn) - \frac{16q}{n^4}$,
1) *$\hat{r} = r$, and*
2) *for all $k = 1, 2, \ldots, q$, $\mathrm{dist}(\boldsymbol{x}_k, \hat{\boldsymbol{x}}_k)^2 \leq c\varepsilon\|\boldsymbol{x}_k\|_2^2$.*

When the goal is to only recover $\boldsymbol{U}$ with subspace error, $\mathrm{SE}(\hat{\boldsymbol{U}}, \boldsymbol{U}) := \|(\boldsymbol{I} - \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}')\boldsymbol{U}\|_2$, at most ε (and not the $\boldsymbol{x}_k$'s), the required lower bounds on m can be significantly relaxed.

Corollary 2. *In the setting of Theorem 1, if $\tilde{m} = 0$, $m \geq \frac{c\kappa^2 \cdot r^2 \log n}{\varepsilon^2}$ and $mq \geq \frac{c\rho^2\kappa^2 \cdot nr^2}{\varepsilon^2}$, then, with same probability, $\mathrm{SE}(\hat{\boldsymbol{U}}, \boldsymbol{U}) \leq c\varepsilon$.*

Discussion. The above results show that, if the goal is to only initialize $\boldsymbol{U}$ with subspace recovery error below a fixed level, say $\varepsilon = 1/4$, then a total of $mq = cnr^2/\varepsilon^2 = 16cnr^2$ iid Gaussian measurements suffice with high probability (whp). When r is small, nr^2 is only slightly larger than nr which is the minimum required by any technique to recover the span of $\boldsymbol{U}$. If the goal is to also initialize the $\boldsymbol{x}_k$'s with normalized error below say $\varepsilon = 1/4$, then more measurements are needed but still significantly fewer than $m = cn$, e.g., if $r \leq c\log n$ and $q = cn$, then $m + \tilde{m} = 16c\sqrt{n}$ suffices.



(a) $m = 8n$, $q = 1000$ (b) $m = 0.8n$, $q = 1000$ (c) $m = 0.6n$, $q = 1000$

Fig. 1: *Plot of reconstruction error,* $\mathrm{NormErr}(\hat{\boldsymbol{X}}^t, \boldsymbol{X})$*, as a function of time taken until iteration* t*. We used complex Gaussian measurement vectors,* $n = 100$, $r = 2$ *and* $q = 1000$ *to generate the data. In Fig. 1a, we show the power of the proposed LRPR initialization. When this is used to initialize TWF (LRPR+TWF), it converges much more quickly than the original TWF. This figure used* $m = 8n$ *(enough measurements for basic TWF to also converge). If* m *is reduced to* $m = 0.8n$ *measurements (Fig. 1b), neither of TWF or LRPR+TWF converge. TWFproj also does not converge because its initialization error is larger. In contrast, both of LRPR1 and LRPR2 converge. LRPR1 is significantly faster than LRPR2 because its per iteration cost is lower. If* m *is reduced further to* $m = 0.6n$ *(Fig. 1c), then LRPR1 does not converge whereas LRPR2 still does.*

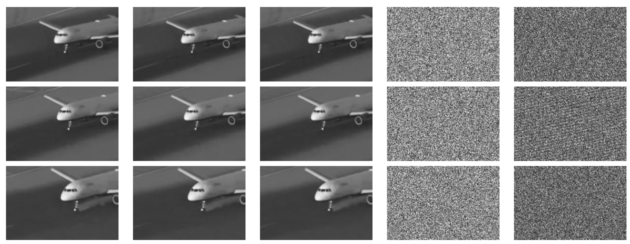


Fig. 2: *This figure shows the power of the proposed methods for recovering a real video from coded diffraction pattern (CDP) measurements. First column: frames 1, 50 and 104, of the original plane video. Next three columns: frames recovered using the various methods from* $m = 3n$ *CDP measurements. Both LRPR1 and LRPR2 significantly outperform TWFproj and TWF. This experiment is inspired by an analogous experiment for recovering a regular camera image from CDP measurements reported in [5, Fig. 2]. While this is not a real practical application since the video used is a regular camera video of a moving airplane, this experiment is done only to illustrate the fact that many real image sequences are indeed approximately low-rank; and that our algorithm has significant advantage over single vector PR methods for jointly recovering this approximately low-rank video.*

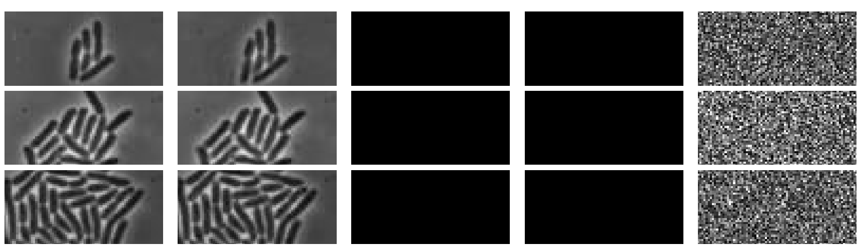


Fig. 3: *This figure shows the power of LRPR2 for recovering a real video from coded diffraction pattern (CDP) measurements. First column: frames 2, 53 and 102, of the original bacteria video. Next three columns: frames recovered using the various methods from* $m = 3n$ *CDP measurements.*

References

[1] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM J. Imaging Sci.*, vol. 6, no. 1, pp. 199–225, 2013.
[2] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Comm. Pure Appl. Math.*, 2013.
[3] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013, pp. 2796–2804.
[4] 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.
[5] 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.
[6] H. Zhang, Y. Zhou, Y. Liang, and Y. Chi, "Reshaped wirtinger flow and incremental algorithm for solving quadratic system of equations," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2016; also at arXiv:1605.07719.
[7] G. Wang, G. B. Giannakis, and Y. C. Eldar, "Solving systems of random quadratic equations via truncated amplitude flow," *arXiv preprint arXiv:1605.08285*, 2016.
[8] 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.
[9] 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.
[10] N. Vaswani, S. Nayer, and Y. C. Eldar, "Low rank phase retrieval," *Revised and Resubmitted to IEEE Trans. Sig. Proc.*, 2016, also at arXiV:1608.04141.
[11] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *Compressed sensing*, pp. 210–268, 2012.