LOW-RANK PHASE RETRIEVAL WITH STRUCTURED TENSOR MODELS

Soo Min Kwon, Xin Li, Anand D. Sarwate

Rutgers, The State University of New Jersey

ABSTRACT

We study the low-rank phase retrieval problem, where the objective is to recover a sequence of signals (typically images) given the magnitude of linear measurements of those signals. Existing solutions involve recovering a matrix constructed by vectorizing and stacking each image. These solutions model this matrix to be low-rank and leverage the low-rank property to decrease the sample complexity required for accurate recovery. However, when the number of available measurements is more limited, these low-rank matrix models can often fail. We propose an algorithm called Tucker-Structured Phase Retrieval (TSPR) that models the sequence of images as a tensor rather than a matrix that we factorize using the Tucker decomposition. This factorization reduces the number of parameters that need to be estimated, allowing for a more accurate reconstruction. We demonstrate the effectiveness of our approach on real video datasets under several different measurement models.

Index Terms— phase retrieval, tensor recovery, non-convex optimization

1. INTRODUCTION

Phase retrieval, or quadratic sensing, is a problem that arises from a wide range of imaging domains such as X-ray crystallography [1], Fourier ptychography [2, 3], and astronomy [4]. In each of these domains, the measurement acquisition process generally involves an optical sensor that captures the diffracted patterns of the object of interest. However, the physical limitations of these sensors only allow us to observe the intensities (or magnitudes) of these patterns. The objective of phase retrieval is then to recover this object $\mathbf{x} \in \mathbb{C}^n$, given a sampling matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$ and measurements $\mathbf{y} \in \mathbb{R}^m$, where $\mathbf{y} = |\mathbf{A}^*\mathbf{x}|$ (or equivalently $\mathbf{y} = |\mathbf{A}^*\mathbf{x}|^2$) and $*$ represents the Hermitian (or conjugate) transpose. The importance of solving the phase retrieval problem in these imaging domains have led to many convex and non-convex solutions [5–10]. However, the theoretical guarantees of all existing methods require the system to be over-determined (i.e. $m \gg n$). This requirement, which is also considered the bottleneck of phase retrieval, mainly comes from the non-convex nature of the problem. In order to converge to the optimal solution, one needs enough samples to guarantee that the initial estimate of the signal is close to the true signal with high probability. This initial estimation step is called spectral initialization, where the term "spectral" comes from the use eigenvectors (or singular vectors) of properly designed matrices from data [11]. This step has been shown to be essential for solving the phase retrieval problem, and many variants of this step have been proposed in the literature.

Recently, there has been a surge of interest in solving the *low-rank phase retrieval* problem [12–15]. This problem can be viewed as a dynamic extension of the standard phase retrieval problem, where the objective is to recover a matrix of vectorized images rather than a single image. Formally, we want to estimate a low-rank matrix $\mathbf{X} \in \mathbb{C}^{n\times q}$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_q]$, given sampling matrices $\mathbf{A}_k \in \mathbb{C}^{n\times m}$ and measurements $\mathbf{y}_k = |\mathbf{A}_k^*\mathbf{x}_k|$, $k = 1, \ldots, q$. Note that in this problem formulation, we assume that there is a separate, independent set of sampling matrices $\mathbf{A}_k$ for each signal $\mathbf{x}_k$. Unlike the phase retrieval problem, this problem has several solutions that have strong theoretical guarantees even for the under-determined setting (i.e. $m \ll n$). These algorithms exploit the low-rank property of the matrix $\mathbf{X}$ with the extra set of sampling matrices in order to naturally reduce the sample complexity. However, our empirical results suggest that there is perhaps a gap between theory and practice, and that these solutions fail to accurately recover the images in the under-determined setting. In fact, in these settings, we observe that these algorithms often do not converge.

In this paper, we propose an algorithm called Tucker-Structured Phase Retrieval (TSPR) that models the sequence of images as a tensor rather than a matrix. With a tensor model, we can decompose the tensor using the Tucker decomposition [16], allowing us to estimate fewer parameters than the matrix counterpart. In the literature, it has been shown that this idea of modelling the parameters as a tensor have been effective in solving many other statistical estimation problems [17–19]. The reduction in the number of parameters also decreases the number of degrees of freedom, suggesting that recovery is possible with a smaller sample complexity. We empirically show that this idea also works for low-rank phase retrieval, on real video datasets with measurements generated from real and complex Gaussian vectors

This work was supported by the US NSF under award CCF-1910110. A longer version of this paper and supplementary materials are available online: https://gitlab.com/sarwate_lab

and coded diffraction patterns. Our results show that in all of these measurement settings, TSPR outperforms existing algorithms in both the under and over-determined regimes.

2. UNSTRUCTURED LOW-RANK PHASE RETRIEVAL

There are several provably efficient algorithms for solving the low-rank phase retrieval problem that vectorize each image and recover a low-rank matrix [12, 13, 15, 20]. We call such methods "unstructured" because they assume no structure in the images. Recently, Nayer et al. proposed AltMinLowRaP [13] which improves AltMinTrunc [12, 15] to solve the unstructured low-phase retrieval problem. AltMinLowRaP alternately minimizes the factor matrices $\mathbf{U} \in \mathbb{C}^{n\times r}$ and $\mathbf{B} \in \mathbb{C}^{q\times r}$ of the low-rank matrix $\mathbf{X} = \mathbf{U}\mathbf{B}^*$. Updating the factor matrix $\mathbf{U}$ consisted of minimizing the objective function

$$\underset{\mathbf{U}}{\operatorname{argmin}} \sum_k \|\mathbf{C}_k\mathbf{y}_k - \mathbf{A}_k^*\mathbf{U}\mathbf{b}_k\|_2^2, \tag{1}$$

where $\mathbf{b}_k$ is the k-th row of the matrix $\mathbf{B}$ and $\mathbf{C}_k$ is a diagonal phase matrix. Note that this objective function sums over all of the columns in $\mathbf{X}$, as the k-th column of $\mathbf{X}$ can be written as $\mathbf{x}_k = \mathbf{U}\mathbf{b}_k$. The intuition behind this summation can be viewed as each of the vectorized images $\mathbf{x}_k$ differing by $\mathbf{b}_k$, while sharing the same span$(\mathbf{U})$. Optimizing for $\mathbf{U}$ involved minimizing this objective function using conjugate gradient least squares (CGLS) while keeping $\mathbf{b}_k$ fixed. The factor matrix $\mathbf{B}$ was initialized and updated by solving an r-dimensional noisy phase retrieval problem for each row $\mathbf{B}$, $\mathbf{b}_k$. To see this, we can rewrite each of the measurements as

$$y_{i,k} = |\langle \mathbf{a}_{i,k}, \mathbf{x}_k\rangle| = |\langle \mathbf{a}_{i,k}, \mathbf{U}\mathbf{b}_k\rangle| = |\langle \mathbf{U}^*\mathbf{a}_{i,k}, \mathbf{b}_k\rangle|. \tag{2}$$

Given an estimate of $\mathbf{U}$, we can solve for each $\mathbf{b}_k$ using any phase retrieval method, such as Reshaped Wirtinger Flow (RWF). Thus, AltMinLowRaP runs RWF q times (once for each image) to estimate $\mathbf{b}_k$ given the sampling matrix $\mathbf{U}^*\mathbf{a}_{i,k}$.

Due to the non-convex nature of this problem, the factor matrix $\mathbf{U}$ was initialized via a spectral method. The matrix $\mathbf{U}$ was initialized by taking the top r eigenvectors of the surrogate matrix

$$\mathbf{Y} = \frac{1}{mq}\sum_{i=1}^{m}\sum_{k=1}^{q} y_{i,k}^2\mathbf{a}_{i,k}\mathbf{a}_{i,k}^*\mathbf{1}_{\{y_{i,k}^2 \le \frac{\alpha^2}{mq}\sum_{u,v} y_{u,v}^2\}}, \tag{3}$$

for some trimming threshold α. The intuition behind this matrix is that given enough samples, the expectation of this matrix is equivalent to

$$\mathbb{E}[y_{i,k}\mathbf{a}_{i,k}\mathbf{a}_{i,k}^*] = 2\mathbf{x}_k\mathbf{x}_k^* + \|\mathbf{x}_k\|^2\mathbf{I}. \tag{4}$$

Thus, the subspace spanned by the top r eigenvectors of $\mathbf{Y}$ can recover exactly $\mathbf{U}$. The double summation over the measurements and samples in the surrogate matrix and truncation is what guaranteed AltMinLowRaP a smaller sample complexity over existing methods. Our algorithm is an empirical improvement over AltMinLowRaP. Although we do not yet have a theoretical analysis of the sample complexity, our results suggest that our algorithm can work better in practice.

3. TUCKER-STRUCTURED PHASE RETRIEVAL

Our algorithm models the sequence of q images as a tensor by reshaping and stacking each of the vectorized images from $\mathbf{x}_k \in \mathbb{C}^n$ into $\mathbf{X}_k \in \mathbb{C}^{n_1\times n_2}$, where $n = n_1n_2$. The objective of TSPR is to recover this tensor $\underline{\mathbf{X}} \in \mathbb{C}^{n_1\times n_2\times q}$, where $\underline{\mathbf{X}}$ can be factorized using the Tucker decomposition written as

$$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times_1 \mathbf{D} \times_2 \mathbf{E} \times_3 \mathbf{F}, \tag{5}$$

where $\underline{\mathbf{G}} \in \mathbb{C}^{r_1\times r_2\times r_3}$ is the core tensor and $\mathbf{D} \in \mathbb{C}^{n_1\times r_1}$, $\mathbf{E} \in \mathbb{C}^{n_2\times r_2}$, and $\mathbf{F} \in \mathbb{C}^{q\times r_3}$ are the factor matrices. Here, r_1 and r_2 refer to the ranks of the frontal slices of the tensor (an image), whereas r_3 refers to the temporal rank that corresponds to the "rank" in the standard model which vectorizes the images. We want to solve for these factors by first initializing them via a spectral method and then estimate using alternating minimization and CGLS.

Spectral Initialization: The idea behind our spectral initialization step is to construct a tensor that is close to $\underline{\mathbf{X}}$ with high probability. Upon constructing this tensor, we can use higher-order SVD (HOSVD) [21] to initialize our core tensor and factor matrices. We adopt the initialization technique of Truncated Wirtinger Flow (TWF) [9] to obtain an initial estimate of the vectorized image $\mathbf{x}_k$. After finding an initial estimate for each image, we can reshape $\mathbf{x}_k$ into its original dimensions and stack them to create the initial tensor. This initialization step is outlined in Algorithm 1.

Alternating Minimization: Upon initialization, we can alternately update the core tensor and each factor matrix using CGLS and RWF. Recall that in AltMinLowRaP, we minimized an objective function that was formed by plugging in $\mathbf{x}_k = \mathbf{U}\mathbf{b}_k$. Similarly, we can minimize the same function, but by rewriting $\mathbf{x}_k$ using our Tucker factors. Specifically, we rewrite each $\mathbf{x}_k$ as $\mathbf{x}_k = (\mathbf{f}_k \otimes \mathbf{E} \otimes \mathbf{D})\text{vec}(\underline{\mathbf{G}})$, where $\mathbf{f}_k$ is the k-th row of the factor matrix $\mathbf{F}$. The reason behind writing $\mathbf{x}_k$ in terms of $\mathbf{f}_k$ is the same reasoning used for the unstructured case – each image $\mathbf{x}_k$ differs by $\mathbf{f}_k$. By plugging in $\mathbf{x}_k$, the update steps of the core tensor $\underline{\mathbf{G}}$ and factor matrices $\mathbf{D}$ and $\mathbf{E}$ consists of minimizing the function

$$\sum_k \|\mathbf{C}_k\mathbf{y}_k - \mathbf{A}_k^*(\mathbf{f}_k \otimes \mathbf{E} \otimes \mathbf{D})\text{vec}(\underline{\mathbf{G}})\|_2^2. \tag{6}$$

To update each row vector $\mathbf{f}_k$, note that we can rewrite $y_{i,k}$ as

$$y_{i,k} = |\langle \mathbf{a}_{i,k}, \mathbf{x}_k\rangle| = |\langle \mathbf{a}_{i,k}, \mathcal{M}_3(\underline{\mathbf{G}})(\mathbf{E}\otimes\mathbf{D})^*\mathbf{f}_k\rangle| \tag{7}$$

$$= |\langle \mathcal{M}_3(\underline{\mathbf{G}})(\mathbf{E}\otimes\mathbf{D})^*\mathbf{a}_{i,k}, \mathbf{f}_k\rangle|, \tag{8}$$

where $\mathcal{M}_k(\underline{\mathbf{G}})$ is the k-th mode matricization of the tensor $\underline{\mathbf{G}}$. With this formulation, updating each $\mathbf{f}_k$ simplifies to solving a noisy r-dimensional phase retrieval problem with sampling matrix $\mathcal{M}_3(\underline{\mathbf{G}})(\mathbf{E} \otimes \mathbf{D})^* \mathbf{a}_{i,k}$. We can use any classical phase retrieval method to solve for $\mathbf{f}_k$, but we use RWF [10] to directly compare the performance of TSPR with AltMinLowRaP. This update step is summarized in Algorithm 2, and the details for implementation are available in the longer version of this paper.

4. NUMERICAL EXPERIMENTS

We compare the performance of TSPR with two closely related algorithms, AltMinTrunc and AltMinLowRaP, using two real video datasets, Mouse and Plane. We consider measurements generated by real Gaussian matrices, complex Gaussian matrices, and coded diffraction patterns (CDP). To quantitatively compare these algorithms, we use the phase-invariant matrix distance [13] defined as

$$\text{mat-dist}^2(\hat{\mathbf{X}}, \mathbf{X}) = \sum_{k=1}^{q} \text{dist}^2(\hat{\mathbf{x}}_k, \mathbf{x}_k), \tag{9}$$

where $\mathbf{X}$ is the true matrix, $\hat{\mathbf{X}}$ is the reconstructed matrix and

$$\text{dist}(\hat{\mathbf{x}}, \mathbf{x}) = \min_{\phi \in [0, 2\pi]} \|\mathbf{x} - e^{\sqrt{-1}\phi} \hat{\mathbf{x}}\|. \tag{10}$$

Some of the results went through a "model correction" step as proposed by Nayer et al. [13]. We provide additional information on this correction step in the longer version of this paper. We also provide a reconstruction of the videos as a supplement and display single frames in this paper.

Experiments with the Mouse Dataset: The mouse dataset is a video of a mouse moving slowly towards a camera, provided by Nayer et al. [13]. The mouse video consisted of 90 frames, where each frame was downsized to be of dimensions 40×80. Upon constructing the tensor $\underline{\mathbf{X}} \in \mathbb{C}^{40 \times 80 \times 90}$, we generated measurements according to the model

$$\mathbf{y}_k = |\mathbf{A}_k^* \text{vec}(\mathbf{X}_k)|, \; k = 1, \ldots, q, \tag{11}$$

where each column of $\mathbf{A}_k$ was drawn either from $\mathbf{a}_{i,k} \sim \mathcal{N}(0, \mathbf{I})$ (real Gaussian distribution) or $\mathbf{a}_{i,k} \sim \mathcal{CN}(0, \mathbf{I})$ (circularly complex Gaussian distribution). We compare the three algorithms in two under-determined settings under these measurements. The numerical results are recorded in Table 1 with two of the reconstructed frames shown in Figure 1. In Table 1, we can see that TSPR outperformed the other two algorithms in both under-determined settings by estimating less parameters. In fact, we observe that for two different ranks, AltMinTrunc did not converge and had a resulting error that was significantly higher than the others. These values were obtained by running $T = 20$ iterations of the total algorithm and $T_{RWF} = 25$ where applicable. For the trimming threshold, we used a value of $\alpha = 3$, as suggested in TWF [9].

Algorithm 1 TSPR Initialization

Input: Observations: $\{y_{i,k} \,|\, 1 \le i \le m, 1 \le k \le q\}$, Sampling vectors: $\{\mathbf{a}_{i,k} \,|\, 1 \le i \le m, 1 \le k \le q\}$, Trimming threshold: α, ranks $= [r_1, r_2, r_3]$

for $k = 1, \ldots, q$ **do**

Compute $\lambda_k = \sqrt{\frac{1}{m} \sum_{i=1}^{m} y_{i,k}}$.

Compute $\mathbf{z}_k$ as leading eigenvector of

$$\mathbf{Y}_k = \sum_{i=1}^{m} y_{i,k}^2 \mathbf{a}_{i,k} \mathbf{a}_{i,k}^* \mathbf{1}_{\{|y_{i,k}|^2 \le \alpha^2 \lambda_k^2\}}$$

Compute $\mathbf{x}_k = \sqrt{\frac{mn}{\sum_{i=1}^{m} \|\mathbf{a}_{i,k}\|_2^2}} \lambda_k \mathbf{z}_k$.

Reshape $\mathbf{x}_k \in \mathbb{C}^n$ into $\mathbf{X}_k \in \mathbb{C}^{n_1 \times n_2}$.

end

Stack tensor into $\underline{\mathbf{X}} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_q]$.

Initialize factors $\mathbf{D}^0, \mathbf{E}^0, \mathbf{F}^0, \underline{\mathbf{G}}^0 = \text{HOSVD}(\underline{\mathbf{X}}, \text{ranks})$

Output: $\mathbf{D}^0, \mathbf{E}^0, \mathbf{F}^0, \underline{\mathbf{G}}^0$

Algorithm 2 Tucker-Structured Phase Retrieval (TSPR)

Input: Observations: $\{y_{i,k} \,|\, 1 \le i \le m, 1 \le k \le q\}$, Sampling vectors: $\{\mathbf{a}_{i,k} \,|\, 1 \le i \le m, 1 \le k \le q\}$, Initial factors: $\mathbf{D}^0, \mathbf{E}^0, \mathbf{F}^0, \underline{\mathbf{G}}^0$, Iterations T, RWF Iterations T_{RWF}

for $t = 1, \ldots, T$ **do**

for $k = 1, \ldots, q$ **do**

Update $\mathbf{f}_k^{t+1} = \text{RWF}([\mathbf{D}^t, \mathbf{E}^t, \underline{\mathbf{G}}^t, \mathbf{A}_k^*], \mathbf{y}_k, T_{RWF})$

Compute $\mathbf{X}_k^{t+1} = (\mathbf{f}_k^{t+1} \otimes \mathbf{E}^t \otimes \mathbf{D}^t) \text{vec}(\underline{\mathbf{G}}^t)$

Update phases $\mathbf{C}_k^{t+1} = \text{Diag}(\text{Phase}(\mathbf{A}_k^* \text{vec}(\mathbf{X}_k^{t+1})))$

end

Update $\mathbf{D}^{t+1}$, $\mathbf{E}^{t+1}$, $\underline{\mathbf{G}}^{t+1}$ by minimizing (6)

end

Reconstruct tensor $\underline{\mathbf{X}}^T = \underline{\mathbf{G}}^T \times_1 \mathbf{D}^T \times_2 \mathbf{E}^T \times_3 \mathbf{F}^T$

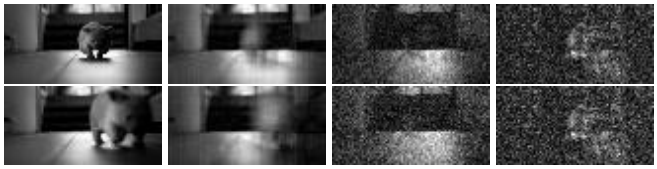
Output: $\underline{\mathbf{X}}^T$

The ranks were generally chosen by trial and error, and the results did not go through a model correction step, as it seemed to increase the errors numerically. We would also like to note that although TSPR yielded a lower numerical reconstruction error, we can see in Figure 1 that the reconstructed image is still not as clear as the original image. This is an intrinsic tradeoff of the Tucker model, as each frame may not be exactly low-rank. We want to choose the ranks corresponding to the image dimensions (i.e. r_1, r_2) to be small so that we can get convergence up to some modelling error, but not too small such that the reconstructed images are unclear. Based on our experiments, we observed that for ranks r_1 and r_2, using ranks slightly less than half of the dimensions of image (i.e. $r_1 < 0.5n_1$ and $r_2 < 0.5n_2$) worked well, whereas for r_3 (or r in the matrix model), we can be more conservative in our choices and choose a value much smaller.

Experiments with the Plane Dataset: The plane dataset is

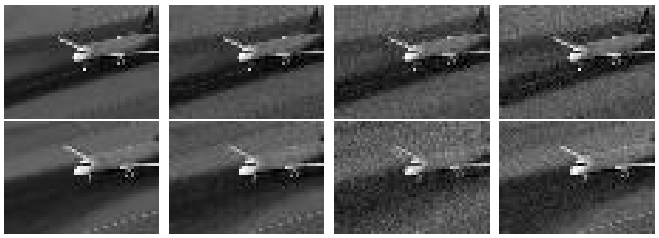
Experiment	Samples	# of Parameters	Algorithm	Rank	Distance
Mouse (Real)	$m = 0.25n$	5750	TSPR	$r = [20, 25, 5]$	2.851
		16450	AltMinLowRaP	$r = 5$	6.175
		16450	AltMinTrunc	$r = 5$	7.277
Mouse (Complex)	$m = 0.75n$	5750	TSPR	$r = [20, 25, 5]$	1.217
		8700		$r = [20, 25, 10]$	1.170
		16450	AltMinLowRaP	$r = 5$	4.379
		32900		$r = 10$	3.435
		16450	AltMinTrunc	$r = 5$	78.118
		32900		$r = 10$	77.319
Plane (CDP)	$m = 2n$	5600	TSPR	$r = [15, 20, 10]$	0.437
		8075		$r = [20, 25, 10]$	0.571
		14525		$r = [30, 35, 10]$	1.008
		22900	AltMinLowRaP	$r = 10$	0.869
		22900	AltMinTrunc	$r = 10$	0.894

Table 1: Results for the experiments with the Mouse and Plane datasets. The value n refers to the dimensions of $\mathbf{x}_k$ and m refers to the number of measurements generated for each $\mathbf{x}_k$. The # of parameters value refers to the total number of parameters that need to be solved for all images $\mathbf{x}_k$. The distance metric is the phase-invariant distance defined in equation (9).



Original TSPR AltMinLowRaP AltMinTrunc

Fig. 1: Results from recovering a video of a moving mouse from complex Gaussian measurements. Rows 1 and 2: reconstructed images of frames 60 and 70, respectively.



Original TSPR AltMinLowRaP AltMinTrunc

Fig. 2: Results from recovering a video of a plane from CDP measurements. Rows 1 and 2: reconstructed images of frames 10 and 80, respectively.

a video of a plane slowly landing on a runway, also provided by Nayer et al. [13]. The plane video consisted of 90 frames, where each frame was downsized to be of dimensions 40×55 for efficiency. Upon constructing the tensor $\underline{\mathbf{X}} \in \mathbb{C}^{40\times55\times90}$, we generated measurements according to the CDP model

$$\mathbf{y}_{l,k} = |\tilde{\mathbf{F}}\mathbf{M}_l \text{vec}(\mathbf{X}_k)|,\ l = 1, \dots, L,\ k = 1, \dots q, \quad (12)$$

where $\tilde{\mathbf{F}}$ is the discrete Fourier transform (DFT) matrix and $\mathbf{M}$ is a diagonal mask matrix with elements drawn randomly from $\{1, -1, j, -j\}$. Since the CDP model can only generate measurements $m = Ln$ for each image for some integer L, the objective of this experiment was to show the effectiveness of TSPR in the over-determined setting. Upon running all three algorithms with the same parameters as the Mouse dataset, each result went through a model correction step. In Figure 2, we see that while all three algorithms can visually reconstruct the frames of this video, Table 1 shows that the errors for TSPR are lower. However, the errors are only lower for certain values of the Tucker rank. This is most likely because as these ranks increase, the total number of parameters slowly converge to that of the unstructured methods, making recovery more difficult.

5. CONCLUSION

In this paper, we showed that by modelling the sequence of images as a tensor, we can obtain a more accurate reconstruction in both the under and over-sampled regimes. Our algorithm, TSPR, adopted a mixture of optimization techniques from AltMinLowRaP and Truncated Wirtinger Flow to improve upon existing methods. TSPR involved a spectral initialization method that used higher-order SVD with alternating minimization via conjugate gradient least squares. Currently, TSPR lacks the theoretical guarantees in comparison to unstructured solutions. One important avenue for future research can be to extend our algorithm but with theoretical guarantees on the sample complexity required for accurate recovery. Our results show that there *exist* Tucker-structured models with better performance; we believe that perhaps finding a more principled approach for choosing these ranks is an important challenge for future work.

6. REFERENCES

[1] Rick P. Millane, "Phase retrieval in crystallography and optics," *Journal of The Optical Society of America A-optics Image Science and Vision*, vol. 7, no. 3, pp. 394–411, 1990.

[2] Gauri Jagatap, Zhengyu Chen, Seyedehsara Nayer, Chinmay Hegde, and Namrata Vaswani, "Sample efficient Fourier ptychography for structured data," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 344–357, 2020.

[3] Jason Holloway, M. Salman Asif, Manoj Kumar Sharma, Nathan Matsuda, Roarke Horstmeyer, Oliver Cossairt, and Ashok Veeraraghavan, "Toward long-distance subdiffraction imaging using coherent camera arrays," *IEEE Transactions on Computational Imaging*, vol. 2, no. 3, pp. 251–265, 2016.

[4] Mark D. Butala, Richard A. Frazin, Yuguo Chen, and Farzad Kamalabadi, "A monte carlo technique for large-scale dynamic tomography," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, pp. 1217–1220, 2007.

[5] Emmanuel J. Candès, Yonina C. Eldar, Thomas Strohmer, and Vladislav Voroninski, "Phase retrieval via matrix completion," *SIAM Review*, vol. 57, no. 2, pp. 225–251, 2015.

[6] Emmanuel J. Candès, Thomas Strohmer, and Vladislav Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, 2013.

[7] Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi, "Phase retrieval using alternating minimization," *IEEE Transactions on Signal Processing*, vol. 63, 2013.

[8] Emmanuel Candès, Xiaodong Li, and Mahdi Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, 2014.

[9] Yuxin Chen and Emmanuel Candès, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[10] Huishuai Zhang, Yingbin Liang, and Yuejie Chi, "A nonconvex approach for phase retrieval: Reshaped wirtinger flow and incremental algorithms," *Journal of Machine Learning Research*, vol. 18, no. 141, pp. 1–35, 2017.

[11] Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma, "Spectral methods for data science: A statistical perspective," Tech. Rep. arXiv:2012.08496v2 [stat.ML], arXiv, 2021.

[12] N. Vaswani, Seyedehsara Nayer, and Yonina C. Eldar, "Low-rank phase retrieval," *IEEE Transactions on Signal Processing*, vol. 65, pp. 4059–4074, 2017.

[13] Seyedehsara Nayer, Praneeth Narayanamurthy, and Namrata Vaswani, "Provable low rank phase retrieval," *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5875–5903, 2020.

[14] Zhengyu Chen, Gauri Jagatap, Seyedehsara Nayer, Chinmay Hegde, and Namrata Vaswani, "Low rank Fourier ptychography," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6538–6542, 2018.

[15] Seyedehsara Nayer, Namrata Vaswani, and Yonina C. Eldar, "Low rank phase retrieval," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4446–4450, 2017.

[16] Tamara Kolda and Brett W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, pp. 455–500, 2009.

[17] Mohsen Ghassemi, Zahra Shakeri, Anand D. Sarwate, and Waheed U. Bajwa, "Learning mixtures of separable dictionaries for tensor data: Analysis and algorithms," *IEEE Transactions on Signal Processing*, vol. 68, no. 1, pp. 33–48, 2020.

[18] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li, "Tucker tensor regression and neuroimaging analysis," *Statistics in Biosciences*, vol. 10, no. 3, pp. 520–545, 2018.

[19] Anru R Zhang, Yuetian Luo, Garvesh Raskutti, and Ming Yuan, "Islet: Fast and optimal low-rank tensor regression via importance sketching," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 2, pp. 444–479, 2020.

[20] Kaihui Liu, Jiayi Wang, Zhengli Xing, Linxiao Yang, and Jun Fang, "Low-rank phase retrieval via variational bayesian learning," *IEEE Access*, vol. 7, pp. 5642–5648, 2019.

[21] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.