Towards Sample-Optimal Methods for Solving Random Quadratic Equations with Structure

Gauri Jagatap and Chinmay Hegde
Electrical and Computer Engineering Department
Iowa State University, Ames, IA, USA 50011

***Abstract*—We consider the problem of estimating a structured high-dimensional parameter vector using random Gaussian quadratic samples. This problem is a generalization of the classical problem of phase retrieval and impacts numerous problems in computational imaging. We provide a generic algorithm based on alternating minimization that, if properly initialized, achieves information-theoretically optimal sample complexity. In essence, we show that solving a system of random quadratic equations with structural constraints is (nearly) as easy as solving the corresponding linear system with the same constraints, if a proper initial guess of the solution is available. As an immediate consequence, our approach improves upon the best known existing sample complexity results for phase retrieval (structured or otherwise). We support our theory via several numerical experiments.**

A full version of this paper is accessible at: https://gaurijagatap.github.io/assets/ISIT18.pdf

I. Introduction

Motivation: Our focus in this paper is the following constrained estimation problem. An unknown vector of parameters, $\mathbf{x}^* \in \mathbb{R}^n$, is observed (or measured) to yield observations $\mathbf{y} \in \mathbb{R}^m$ of the form:

$$y_i = |\langle \mathbf{a}_i, \mathbf{x}^* \rangle|^p, \quad i = [m], \quad \text{s.t.} \quad \mathbf{x}^* \in \mathcal{M}_s \qquad (1)$$

where $\mathcal{M}_s \subset \mathbb{R}^n$ is a *model* set that reflects the structural constraints on $\mathbf{x}^*$. We adopt the familiar setting of underdetermined Gaussian observations, $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_i \dots \mathbf{a}_m]^\top \in \mathbb{R}^{m \times n}$ with $m < n$. The task is to recover an estimate of $\mathbf{x}^*$ from either absolute-value ($p = 1$) or quadratic ($p = 2$) measurements $\mathbf{y}$[1].

An important application of the aforementioned setup is the classical signal processing problem of *phase retrieval*. Here, the measurements correspond to the magnitudes of complex 2D Fourier (or Short Time Fourier) transform coefficients. The sensing apparatus is incapable of detecting phase of the complex light-field reflected or transmitted from the illuminated object source. This necessitates a phase recovery strategy, and proposed solution approaches been explored dating back to the 1970s via several works [1], [2]. Recent, renewed interest by the statistical learning community in this problem has focused on the case of Gaussian observations, and have spawned several algorithms which are efficient as well as asymptotically (near) optimal [3]–[6]. However, even in the best case, one requires $m > 2n - 1$ [6] measurements, and in the case of high dimension n, the sample complexity m can be very large.

Similarly, polynomial neural networks enable applications such as classification, where the activation functions are *quadratic* mappings [7]. The problem in (1) then corresponds to *learning* weights of a single neuron, $\mathbf{x}^*$, for m Gaussian distributed training examples $\mathbf{a}_i$, and y_i being the corresponding output labels. The task is to design efficient algorithms for *learning* weights using fewer training samples.

To reduce sample complexity of such problems, several works introduce *sparsity* assumptions. Sparsity has been used to great advantage in compressive sensing and streaming algorithms [8], [9], and establish an information theoretically optimal [10] requirement of $\mathcal{O}\left(s \log \frac{n}{s}\right)$ samples for stable recovery of $\mathbf{x}^*$ from linear measurements. Sparsity assumptions for inverting quadratic (or magnitude-only) equations of the form (1) has similarly helped lower computational and memory requirements [5], [11]–[13]. Specifically, several papers consider the problem in (1), where $\mathcal{M}_s$ represents all s-sparse vectors $\mathbf{x}^*$, with the assumption of a Gaussian observation framework [14]–[16]. In previous work [16], [17], we have proposed a sparse phase retrieval algorithm called CoPRAM, which is linearly convergent and improves upon all other algorithms, obtaining a Gaussian sample complexity of $\mathcal{O}\left(s^2 \log n\right)$ in general ($\mathcal{O}\left(s \log n\right)$ if power-law decaying sparse signals are considered [17]).

A natural extension of sparsity is the notion of *structured sparsity*. Several works in compressed sensing and statistical learning have utilized various structures such as blocks, clusters, and trees [18], [19]. Block structures in the context of sparse phase retrieval have been studied in [16]. Tree structures are popularly found in applications where sparsity is considered in wavelet basis [20]. While the impact of structured sparsity has been studied thoroughly for the case of linear measurements [21], recovery from quadratic or magnitude-only measurements is relatively less understood.

Our contributions: In this paper, we propose a new algorithmic framework called *Model-based CoPRAM* to solve the problem of phase retrieval of signals with underlying structured sparsity. Our framework is fairly generic and succeeds for parameters belonging to any structured sparsity model (defined formally below). Moreover, we provably show that if the algorithm is properly initialized, then its sample complexity is information-theoretically *optimal*. In particular, we analyze a special instantiation of our framework, called *Tree CoPRAM*,

[1]This work was supported in part by NSF grants CCF-1566281 and CCF-1750920, and a Black and Veatch Faculty Fellowship.

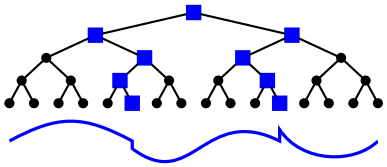


Fig. 1: *A binary wavelet tree for a one-dimensional signal. The squares denote the large wavelet coefficients that arise from the discontinuities in the piecewise smooth signal drawn below. Figure taken from [22].*

which is applicable in the case of rooted s-sparse tree structures for $\mathbf{x}^*$, and demonstrate the superior performance of our method both in theory and numerical simulations. In essence, our contributions imply that solving a system of under-determined quadratic equations under structural constraints is (essentially) as easy as solving the corresponding linear system under the same constraints, provided a good initial guess is available.

Techniques: The algorithmic techniques used in this paper are a combination of two focal points: (i) alternating minimization based sparse signal recovery from phaseless measurements (via our previous work on CoPRAM [16]), (ii) using a structured-sparsity promoting subroutine called ModelApprox (e.g. TreeApprox [22] in the context of tree-structured sparsity), which replaces the standard s-sparse projection rule used to enforce sparsity. Additionally, we also design a novel initialization heuristic, which yields an initial estimate $\mathbf{x^0}$ that is very close to $\mathbf{x}^*$ in practice. Our main theoretical contribution is a generalization of a recent result of [23] to the case where $\mathbf{x}^*$ belongs to a known model set $\mathcal{M}_s$.

II. Background

A. Structured sparsity

We provide some background for the problem formulation in (1). A vector $\mathbf{x}^* \in \mathbb{R}^n$ is said to be s-sparse if it has no more than s non-zero entries. We use $S := \{j | x_j^* \neq 0\}$ to indicate the true support of $\mathbf{x}^*$, such that $|S| \leq s$. The model notation $\mathcal{M}_s$ is introduced as an indicator set comprising of all vectors which follow a given structural constraint, underscored with parameter s. Let $\mathbb{M}_s$ denote the set of all allowable supports $\{S_1 \dots S_i \dots S_N\}$, such that $S_i \subseteq [n]$ and $|S_i| \leq s$, then $\mathcal{M}_s = \{\mathbf{x} \in \mathbb{R}^n | \operatorname{supp}(\mathbf{x}) \in \mathbb{M}_s\}$. As a special case, $\mathcal{M}_s$ can be a model representing all s-sparse rooted tree supports, as illustrated in Figure 1.

B. Model-based CoSaMP

Model-based CoSaMP (Compressive Sensing using Matching Pursuit) [19] is a popular technique to recover structured (for example tree) s-sparse vectors or signals $\tilde{\mathbf{x}}^* \in \mathbb{R}^n$ from linear observations $\tilde{\mathbf{y}} \in \mathbb{R}^m$ of the form:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}^*,$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times n}$ and $m < n$. The sensing matrix $\tilde{\mathbf{A}}$ is required to satisfy model-RIP [19], with constant δ_{M_s}, such that for all $\tilde{\mathbf{x}} \in \mathcal{M}_s$, the following holds:

$$(1 - \delta_{M_s}) \|\tilde{\mathbf{x}}\|_2^2 \leq \left\|\tilde{\mathbf{A}}\tilde{\mathbf{x}}\right\|_2^2 \leq (1 + \delta_{M_s}) \|\tilde{\mathbf{x}}\|_2^2 .$$

This holds trivially, if the entries of $\tilde{\mathbf{A}}$, $\tilde{a}_{ij}$ are distributed according to normal distribution $\mathcal{N}(0, 1/\sqrt{m})$. Model CoSaMP solves the following minimization approximately:

$$\min_{\tilde{\mathbf{x}} \in \mathcal{M}_s} \left\|\tilde{\mathbf{y}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}\right\|_2^2 . \tag{2}$$

Model-based CoSaMP (also referred as ModelCoSaMP), utilizes a model-based approximation stage (referred to as ModelApprox). A specific instantiation of Model-based CoSaMP, Tree CoSaMP employs an exact or approximate *tree projection* subroutine called TreeApprox [22], [24], to ensure that the output of the minimization in (2), $\tilde{\mathbf{x}}^+$, belongs to the model $\mathcal{M}_s$. This approach is largely parameter free and only requires knowledge of signal sparsity s and assumption of tree structure.

C. Phaseless signal recovery

The recovery problem can be expressed by constructing a loss function of the form:

$$\min_{\mathbf{x} \in \mathcal{M}_s} \sum_{i=1}^{m} \left(y_i - |\langle \mathbf{a}_i, \mathbf{x} \rangle|^p\right)^2 . \tag{3}$$

where $p = 1$ or $p = 2$. Gradient descent based approaches popularly use the Wirtinger Flow (which solves the quadratic variant, $p = 2$) [4], [14], [23], [25], [26] and Amplitude Flow (which solves the magnitude-only variant, $p = 1$) [6], [15], [27] approaches, to calculate the explicit gradient of the objective function in (3) composed of either squared or magnitude-only measurements. In this paper, we use the alternating minimization approach [2], with magnitude-only measurements ($p = 1$), by introducing a new variable to represent the missing phase information, hence *linearizing* the problem. We then update the phase variable and signal variable in an *alternating* fashion. In the signal estimation stage, we employ the Model-based CoSaMP algorithm to obtain a structurally sparse vector estimate. To evaluate the distance of the $\mathbf{x}$-estimate from $\mathbf{x}^*$, we introduce the expression $\operatorname{dist}(\mathbf{x}_1, \mathbf{x}_2) := \min(\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \|\mathbf{x}_1 + \mathbf{x}_2\|_2)$ for every $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$. This method is discussed in further detail in Section III-A.

D. Spectral initialization

Non-convex approaches for phase retrieval [4], [5] rely on a spectral initialization technique to ensure that the initial estimate $\mathbf{x^0}$ is within a δ_0-ball radius of the true solution $\mathbf{x}^*$. This is required to establish subsequent convergence of descent based algorithms. For this, one can construct an estimator matrix $\mathbf{M} = \frac{1}{m}\sum_{i=1}^{m} y_i^2 \mathbf{a}_i \mathbf{a}_i^\top$, and use the top left-singular vector of this matrix as an appropriate initialization. Sparse modifications of this strategy involve detecting (partial) support information from the diagonal of the estimator matrix $\mathbf{M}$, by using an approximate projection onto model $\mathcal{M}_s$. This method is discussed in further detail in Section III-B.

III. Algorithm

In this section, we propose a new algorithm for solving the tree sparse phase retrieval problem and analyze its performance. We use a *spectral initialization*, followed by an alternating

Algorithm 1 Model-based CoPRAM

Input: $\mathbf{A}, \mathbf{y}, s, t_0$

1: Compute signal power: $\phi^2 = \frac{1}{m}\sum_{i=1}^{m} y_i^2$.
2: Compute: diag($\mathbf{M}$) := $M_{jj} = \frac{1}{m}\sum_{i=1}^{m} y_i^2 a_{ij}^2$ for $j = [n]$.
3: Set: $\hat{S} \leftarrow$ MODELAPPROX(diag($\mathbf{M}$)).
4: Set: $\mathbf{v}_1 \leftarrow$ top s.v. of $\mathbf{M}_{\hat{S}} = \frac{1}{m}\sum_{i=1}^{m} y_i^2 \mathbf{a}_{i\hat{S}}\mathbf{a}_{i\hat{S}}^T$.
5: Compute: $\mathbf{v} \in \mathbb{R}^n \leftarrow \mathbf{v}_1$ for $\hat{S}$, and $\mathbf{0}$ for $\hat{S}^c$.
6: Compute: $\mathbf{x^0} \leftarrow \phi\mathbf{v}$.

7: **for** $t = 0, \cdots, t_0 - 1$ **do**
8: $\quad \mathbf{p}^{t+1} \leftarrow \text{sign}(\mathbf{A}\mathbf{x}^t)$,
9: $\quad \mathbf{x}^{t+1}$ = MODELCOSAMP($\frac{1}{\sqrt{m}}\mathbf{A}, \frac{1}{\sqrt{m}}\mathbf{p}^{t+1} \circ \mathbf{y}, s, \mathbf{x}^t$).
10: **end for**

Output: $\mathbf{x}^{t_0} \leftarrow \mathbf{x}^t$.

minimization based *descent* approach, similar to our previous work in [16]. Our algorithm is largely parameter-free except for knowledge of the underlying sparsity s. Moreover, the theoretical analysis requires no extra assumptions on the parameter vector, except that its support belongs to a structured sparsity model. We call our algorithm *Model-based CoPRAM*, which generalizes our previous algorithm called CoPRAM (or Compressive Phase Retrieval with Alternating Minimization) [16], [17]. The algorithm can be broken down into three types of update stages: (i) initialization; (ii) phase estimation; and (iii) signal estimation. The full algorithm is presented in pseudo-code form as Algorithm 1. The phase and signal estimation stages are described in detail in the Section III-A. Due to the simplicity of our algorithm, it can easily be extended to a general class of signals defined by any model $\mathcal{M}_s$. In this paper, we focus on the special case where the model $\mathcal{M}_s$ corresponds to tree-sparse vectors in $\mathbb{R}^n$.

A. Convergence of Model-based CoPRAM

This part of the algorithm is described in Lines 7-10 of Algorithm 1. Once we obtain a good enough initial estimate $\mathbf{x^0} \in \mathcal{M}_s$ such that $\text{dist}\left(\mathbf{x^0}, \mathbf{x}^*\right) \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, we construct a method to accurately estimate the true $\mathbf{x}^*$. To achieve this, we adapt the alternating minimization approach from [5]. The observation model in (1) can be restated as follows:

$$\text{sign}\left(\langle \mathbf{a}_i, \mathbf{x}^* \rangle\right) \circ y_i = \langle \mathbf{a}_i, \mathbf{x}^* \rangle ,$$

for all $i = \{1, 2, \ldots m\}$. We denote the *phase vector* $\mathbf{p} \in \mathbb{R}^m$ as a vector that contains the unknown signs of the measurements, i.e., $p_i = \text{sign}\left(\langle \mathbf{a}_i, \mathbf{x} \rangle\right)$ for all $i = \{1, 2, \ldots m\}$. Let $\mathbf{p}^*$ denote the true phase vector and let $\mathcal{P}$ denote the set of all phase vectors, i.e. $\mathcal{P} = \{\mathbf{p} : p_i = \pm 1, \forall i\}$. Then our measurement model gets modified as:

$$\mathbf{p}^* \circ \mathbf{y} = \mathbf{A}\mathbf{x}^*.$$

The loss function in (3) gets modified and is composed of two variables $\mathbf{x}$ and $\mathbf{p}$,

$$\min_{\mathbf{x} \in \mathcal{M}_s, \mathbf{p} \in \mathcal{P}} \left\|\mathbf{A}\mathbf{x} - \mathbf{p} \circ \mathbf{y}\right\|_2 \tag{4}$$

Note that the problem above is *not convex*, because $\mathbf{p} \in \mathcal{P}$ is a set of all vectors with entries constrained to be in $\{-1, 1\}$. Instead, we alternate between estimating $\mathbf{p}$ and $\mathbf{x}$. We perform two estimation steps: (i) if we fix the signal estimate $\mathbf{x}$, then the minimizer $\mathbf{p} \in \mathcal{P}$ is given in closed form as:

$$\mathbf{p} = \text{sign}\left(\mathbf{A}\mathbf{x}\right), \tag{5}$$

(*phase estimation*, Line 8 of Algorithm 1);
(ii) if we fix the phase vector $\mathbf{p}$, the signal vector $\mathbf{x} \in \mathcal{M}_s$ is obtained by solving a (structured) sparse recovery problem,

$$\min_{\mathbf{x} \in \mathcal{M}_s} \frac{1}{\sqrt{m}} \|\mathbf{A}\mathbf{x} - \mathbf{p} \circ \mathbf{y}\|_2, \tag{6}$$

if $m < n$ and $\frac{\mathbf{A}}{\sqrt{m}}$ satisfies the model-RIP as defined above. (*signal estimation*, Line 9 of Algorithm 1).

Here, we employ the Model-based CoSaMP [19] algorithm to (approximately) solve (6). Note that since (6) itself is a non-convex problem, exact minimization can be hard. However, in each signal estimation step, we do not need to explicitly obtain the minimizer; we can still show a sufficient descent criterion by unpackin the analysis of the Model-based CoSaMP algorithm. For analysis reasons, we require that the entries of the input sensing matrix are distributed according to $\mathcal{N}\left(0, 1/\sqrt{m}\right)$. This can be achieved by scaling down the inputs to Model-based CoSaMP: $\mathbf{A}^t, \mathbf{p}^{t+1} \circ \mathbf{y}$ by a factor of $\sqrt{m}$. We also use a "warm start" Model-based CoSaMP routine for the $(t+1)^{th}$ update of $\mathbf{x}$, $\mathbf{x}^{t+1}$, for each iteration where the initial guess of the solution to (6) is given by the current signal estimate $\mathbf{x}^t$.

We now analyze our proposed descent scheme. We obtain the following theoretical result:

Theorem III.1. *Given an initialization* $\mathbf{x^0} \in \mathcal{M}_s$ *satisfying* $\text{dist}\left(\mathbf{x}^0, \mathbf{x}^*\right) \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, *for* $0 < \delta_0 < 1$, *if we have number of Gaussian measurements,*

$$m > C\left(s + \text{card}(\mathbb{M}_{4s})\right),$$

then the iterates $\mathbf{x}^{t+1}$ *of Algorithm 1, satisfy:*

$$\text{dist}\left(\mathbf{x}^{t+1}, \mathbf{x}^*\right) \leq \rho_0 \text{dist}\left(\mathbf{x}^t, \mathbf{x}^*\right), \tag{7}$$

where $\mathbf{x}^t, \mathbf{x}^{t+1}, \mathbf{x}^* \in \mathcal{M}_s$, *and* $0 < \rho_0 < 1$ *is a constant, with probability greater than* $1 - e^{-\gamma m}$, *for positive constant* γ.

Proof sketch: Via an algebraic derivation similar to the one provided in [17], the per-iteration error for the tth iteration of Alg. 1, with L iterations of Model-based CoSaMP, can be derived as:

$$\left\|\mathbf{x}^{t+1} - \mathbf{x}^*\right\|_2 \leq (\rho_1\rho_3)^L \left\|\mathbf{x}^* - \mathbf{x}^t\right\|_2 + \frac{(\rho_1\rho_4 + \rho_2)}{(1 - \rho_1\rho_3)} \left\|E_{ph}\right\|_2, \tag{8}$$

where $\rho_1, \rho_2, \rho_3, \rho_4$ are appropriate constants, and E_{ph} is the error in estimating phase in the tth run of Model-based CoPRAM. The second part of this proof requires a bound on the phase error term $\left\|E_{ph}\right\|_2$, which can be analytically derived as:

$$\|E_{ph}\|_2^2 = \frac{4}{m}\sum_{i=1}^{m}\left(\mathbf{a}_i^\top \mathbf{x}^*\right)^2 \cdot \mathbf{1}_{\{\text{sign}(\mathbf{a}_i\mathbf{x}^t)\,\text{sign}(\mathbf{a}_i\mathbf{x}^*) = -1\}}.$$

We bound this term by invoking Lemma III.2.

Lemma III.2. *As long as the initial estimate is a small distance away from the true signal* $\mathbf{x}^* \in \mathcal{M}_s$, $\operatorname{dist}\left(\mathbf{x^0}, \mathbf{x}^*\right) \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, *and subsequently,* $\operatorname{dist}\left(\mathbf{x}^t, \mathbf{x}^*\right) \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, *where* $\mathbf{x}^t$ *is the* t^{th} *update of Algorithm 1, then the following bound holds,*

$$\frac{4}{m}\sum_{i=1}^{m}\left(\mathbf{a}_i^\top \mathbf{x}^*\right)^2 \cdot \mathbf{1}_{\left\{(\mathbf{a}_i^\top \mathbf{x}^t)(\mathbf{a}_i^\top \mathbf{x}^*)\leq 0\right\}} \leq \rho_5^2 \left\|\mathbf{x}^t - \mathbf{x}^*\right\|_2^2,$$

with probability greater than $1 - e^{-\gamma_2 m}$, *where* γ_2 *is a positive constant, as long as* $m > C\left(s + \operatorname{card}(\mathbb{M}_{4s})\right)$ *and* $\rho_5^2 = 0.0256$.

We therefore achieve a per-step error reduction scheme of the form:

$$\left\|\mathbf{x}^{t+1} - \mathbf{x}^*\right\|_2 \leq \rho_0 \left\|\mathbf{x}^t - \mathbf{x}^*\right\|_2,$$

if the initial estimate $\mathbf{x^0}$ satisfies $\left\|\mathbf{x^0} - \mathbf{x}^*\right\|_2 \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, and this result can be trivially extended to the case where the initial estimate $\mathbf{x^0}$ satisfies $\left\|\mathbf{x^0} + \mathbf{x}^*\right\|_2 \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$, hence giving the convergence criterion of the form (for $\rho_0 < 1$):

$$\operatorname{dist}\left(\mathbf{x}^{t+1}, \mathbf{x}^*\right) \leq \rho_0 \operatorname{dist}\left(\mathbf{x}^t, \mathbf{x}^*\right).$$

The complete proof of Theorem III.1 and Lemma III.2 can be found in Appendix A of the full paper [28]. We present a corollary of Theorem III.1 for *tree* sparse signals.

Corollary III.3. *As a consequence of Theorem III.1, if* $\mathcal{M}_s$ *is a model representing rooted tree sparse signals with sparsity* s, *then Algorithm 1 is linearly convergent and requires a Gaussian sample complexity of* $m > Cs$, *as long as the initialization* $\mathbf{x^0}$ *satisfies* $\operatorname{dist}\left(\mathbf{x}^0, \mathbf{x}^*\right) \leq \delta_0 \left\|\mathbf{x}^*\right\|_2$.

The proof of this corollary can be found in Appendix A of the full version of this paper [28]. Observe that $m = O(s)$ samples are necessary for reconstructing any s-sparse parameter vector even in the linear case (where perfect phase information is available), and therefore Theorem III.1 implies information-theoretic optimality (up to constants) of our proposed approach.

B. Initialization of Model-based CoPRAM

The first stage (Lines 1-6 of Algorithm 1) of Model-based CoPRAM uses a spectral initialization approach, similar to that provided in previous sparse phase retrieval methods [5], [14]–[16], [26]. We construct a *biased* estimator of the squared true signal coefficients, which we call the signal *marginal* matrix:

$$\mathbf{M} = \frac{1}{m}\sum_{i=1}^{m} y_i^2 \mathbf{a}_i \mathbf{a}_i^\top.$$

The jth signal coefficient can be estimated from the the diagonal term $M_{jj} = \frac{1}{m}\sum_{i=1}^{m} y_i^2 a_{ij}^2$, and the set of all M_{jj}'s can be calculated in $\mathcal{O}(mn)$ time. The approximate support estimate $\hat{S}$ can be extracted by performing an exact or approximate tree projection algorithm [22] on the n-dimensional diagonal of the marginal matrix $\mathbf{M}$. From this we obtain the sub-matrix $\mathbf{M}_{\hat{S}}$, whose rows and columns are projected onto $\hat{S}$. This is followed by a spectral technique ([5], [14]–[16]), which extracts the top left singular vector (s.v.) of $\mathbf{M}$ to construct a good initial estimate $\mathbf{x^0}$ (Lines 4-6 of Algorithm 1).

To provide the intuition behind this strategy, we leverage the fact that the diagonal elements of the expectation matrix $\mathbb{E}[\mathbf{M}]$ are given by $\mathbb{E}[M_{jj}] = \|\mathbf{x}^*\|^2 + 2x_j^{*2}$. The signal marginals M_{jj} corresponding to $j \in S$ are larger, in *expectation*, than those corresponding to $j \in S^c$. Therefore the signal marginals M_{jj} serve as a good indicator to extract an approximate support $\hat{S}$ of $\mathbf{x}^*$. We additionally impose structure $\mathcal{M}_s$ to this sparse initial vector, by utilizing an approximate model projection algorithm (such as tree projection [22]) (Line 3 of Algorithm 1). We demonstrate experimentally that this initialization strategy produces a good estimate of $\mathbf{x}^*$. We do not have a full theoretical characterization of the initialization stage, and intend to pursue this in future work.

IV. Experiments

In this section, we present experimental results to demonstrate the empirical advantages of Model-based CoPRAM (specifically, for the tree-sparsity model) over standard sparse phase retrieval algorithms (specifically, CoPRAM [16] which provides the best available empirical performance for the standard case). We consider two different sizes (32×32 and 64×64) of an image of Rice University's Lovett Hall as shown in Figure 2. This image is modeled as sparse in the Haar wavelet basis, with the number of levels of decomposition are chosen to be $\log_2 n$ where n is the number of image pixels.

Fig. 2: *Image considered for simulations, resized to* 32×32 *and* 64×64 *pixels, considered to be sparse in Haar basis.*

The Tree CoPRAM and CoPRAM algorithms were run for the following experimental settings: $n = 1024$ and $n = 4096$. The original image $\hat{\mathbf{x}}$ was *sparsified* by fixing s and picking the top s- wavelet coefficients of $\hat{\mathbf{x}}$. This sparsified image is considered to be the s-sparse tree structured ground truth $\mathbf{x}^*$.

Phase transitions: We demonstrate the superior performance of the Tree CoPRAM algorithm in comparison to CoPRAM, through a series of phase transition graphs and diagrams. In Figure 3, we illustrate two different settings of sparsities for $n = 1024$ dimensional $\mathbf{x}^*$: $s = 10$ and $s = 31$, and compare the performances of CoPRAM and Tree CoPRAM by plotting the variation in the number of measurements m on the horizontal axis and the probability of successful recovery (fraction of trials in which $\|\mathbf{x}^{t_0} - \mathbf{x}^*\|_2 / \|\mathbf{x}^*\|_2 \leq 0.05$). It is clear that far fewer samples are required for successful recovery, when Tree CoPRAM is used instead of CoPRAM.

Figure 4 shows phase transitions for image size n=4096, at different sparsities (s=10,20,31,41,51,61,72,82,92,102) and

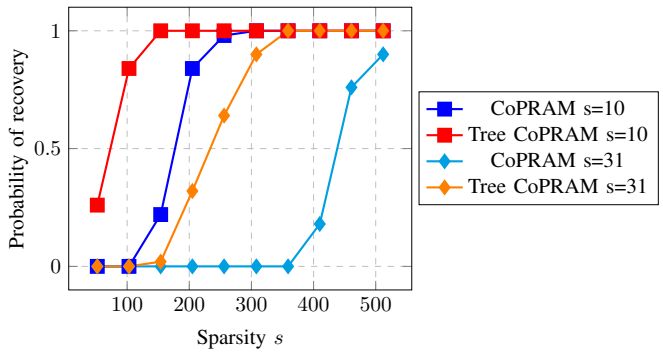
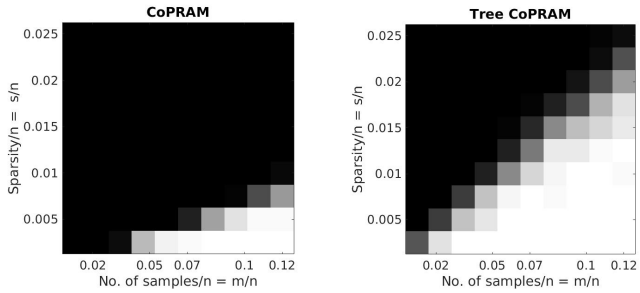


Fig. 3: *Phase transitions for CoPRAM and Tree CoPRAM for sparsities s=10 and s=31 on an n=1024 dimensional signal.*



(a) (b)

Fig. 4: *Phase transition diagrams for (a) CoPRAM (b) Tree CoPRAM on signal of dimension $n = 4096$.*

TABLE I: Average running time in seconds of Tree CoPRAM v/s CoPRAM for n=1024.

Algorithm	s=10,m=308	s=20,m=410	s=20,m=512
CoPRAM	0.0241	0.0455	0.0433
Tree CoPRAM	0.0119	0.0336	0.0302

number of measurements taking (approximately) uniform integer values between m=52 and m=512. It is clear that the phase transition plot of Tree CoPRAM demonstrates better sample complexity w.r.t. CoPRAM.

Running time performance: In our final set of results, the running time performance of Tree CoPRAM w.r.t CoPRAM is tabulated in Table I.

The simulations were run using MATLAB R2017b on a desktop computer with Intel Xeon E5-2620 processor with 12 CPUs at 2.4GHz and 64GB RAM. The comparative performance of CoPRAM w.r.t. other sparse phase retrieval algorithms has been discussed in Section 5 of [16].

References

[1] R. Gerchberg and W. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35(237), 1972.

[2] J. Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

[3] E. Candes, T. Strohmer, and V. Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Comm. Pure Appl. Math.*, 66(8):1241–1274, 2013.

[4] E. Candes, X. Li, and M. Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Trans. Inform. Theory*, 61(4):1985–2007, 2015.

[5] P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 2796–2804, 2013.

[6] G. Wang, G. Giannakis, Y. Saad, and J. Chen. Solving most systems of random quadratic equations. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, 2017.

[7] R. Livni, S. Shalev-Shwartz, and O. Shamir. On the computational efficiency of training neural networks. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 855–863, 2014.

[8] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[9] D. Needell, J. Tropp, and R. Vershynin. Greedy signal recovery review. In *Proc. Asilomar Conf. Sig. Sys. Comput.*, pages 1048–1050. IEEE, 2008.

[10] K. Do Ba, P. Indyk, E. Price, and D. Woodruff. Lower bounds for sparse recovery. In *Proc. ACM Symp. Discrete Alg. (SODA)*, pages 1190–1197, 2010.

[11] M. Moravec, J. Romberg, and R. Baraniuk. Compressive phase retrieval. In *Opt. Eng.+Appl.*, pages 670120–670120. International Society for Optics and Photonics, 2007.

[12] M. Iwen, A. Viswanathan, and Y. Wang. Robust sparse phase retrieval made easy. *Ap. Comp. Har. An.*, 42(1):135–142, 2017.

[13] H. Ohlsson, A. Yang, R. Dong, and S. Sastry. Cprl–an extension of compressive sensing to the phase retrieval problem. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 1367–1375, 2012.

[14] T. Cai, X. Li, and Z. Ma. Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *Ann. Stat.*, 44(5):2221–2251, 2016.

[15] G. Wang, L. Zhang, G. Giannakis, M. Akçakaya, and J. Chen. Sparse phase retrieval via truncated amplitude flow. *IEEE Transactions on Signal Processing*, 66(2):479–491, 2018.

[16] G. Jagatap and C. Hegde. Fast, sample efficient algorithms for structured phase retrieval. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 4924–4934, 2017.

[17] G. Jagatap and C. Hegde. Sample efficient algorithms for recovering structured signals from magnitude-only measurements. *arXiv preprint arXiv:1705.06412*, 2017.

[18] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Royal Stat. Soc. Stat. Meth.*, 68(1):49–67, 2006.

[19] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. Info. Theory*, 56(4):1982–2001, Apr. 2010.

[20] C. Cartis and A. Thompson. An exact tree projection algorithm for wavelets. *IEEE Sig. Proc. Lett.*, 20(11):1026–1029, 2013.

[21] C. Hegde, P. Indyk, and L. Schmidt. Fast algorithms for structured sparsity. *Bulletin of the EATCS*, 1(117):197–228, Oct. 2015.

[22] C. Hegde, P. Indyk, and L. Schmidt. A fast approximation algorithm for tree-sparse recovery. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1842–1846, June 2014.

[23] H. Zhang and Y. Liang. Reshaped wirtinger flow for solving quadratic system of equations. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 2622–2630, 2016.

[24] C. Hegde, P. Indyk, and L. Schmidt. Nearly linear-time model-based compressive sensing. In *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, pages 588–599, July 2014.

[25] E. Candes, X. Li, and M. Soltanolkotabi. Phase retrieval from coded diffraction patterns. *Ap. Comp. Har. An.*, 39(2):277–299, 2015.

[26] Y. Chen and E. Candes. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 739–747, 2015.

[27] G. Wang and G. Giannakis. Solving random systems of quadratic equations via truncated generalized gradient flow. In *Adv. Neural Inf. Proc. Sys. (NIPS)*, pages 568–576, 2016.

[28] G. Jagatap and C. Hegde. Towards sample-optimal methods for solving random quadratic equations with structure, 2018. Full version: https://gaurijagatap.github.io/assets/ISIT18.pdf.