



Unlabeled Sensing Using Rank-One Moment Matrix Completion

Hao Liang

Key Lab of Mathematics Mechanization, AMSS
University of Chinese Academy of Sciences
Beijing, 100190, China
lianghao2020@amss.ac.cn

Manolis C. Tsakiris

Key Lab of Mathematics Mechanization, AMSS
University of Chinese Academy of Sciences
Beijing, 100190, China
manolis@amss.ac.cn

Jingyu Lu

Key Lab of Mathematics Mechanization, AMSS
University of Chinese Academy of Sciences
Beijing, 100190, China
lujingyu@amss.ac.cn

Lihong Zhi

Key Lab of Mathematics Mechanization, AMSS
University of Chinese Academy of Sciences
Beijing, 100190, China
lzhi@mmrc.iss.ac.cn

ABSTRACT

We study the unlabeled sensing problem that aims to solve a linear system of equations $Ax = \pi(y)$ for an unknown permutation π . For a generic matrix A and a generic vector y , we construct a system of polynomial equations whose unique solution satisfies $A\xi^* = \pi(y)$. In particular, ξ^* can be recovered by solving the rank-one moment matrix completion problem. We propose symbolic and numeric algorithms to compute the unique solution. Some numerical experiments are conducted to show the efficiency and robustness of the proposed algorithms.

KEYWORDS

unlabeled sensing, birational morphism, matrix completion, determinantal variety, semidefinite relaxation

ACM Reference Format:

Hao Liang, Jingyu Lu, Manolis C. Tsakiris, and Lihong Zhi. 2024. Unlabeled Sensing Using Rank-One Moment Matrix Completion. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '24)*, July 16–19, 2024, Raleigh, NC, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3666000.3669674>

1 INTRODUCTION

Given a full-rank matrix $A^* \in \mathbb{R}^{m \times n}$ and a vector $y^* \in \mathbb{R}^s$ satisfying $m \geq s > n > 0$, the *unlabeled sensing problem* [64, 65] asks that for an unknown vector $\xi^* \in \mathbb{R}^n$, if one only knows the vector $y^* \in \mathbb{R}^s$ consisting of s shuffled entries of $A^*\xi^*$, whether the vector ξ^* is unique and how to recover it efficiently. This problem emerges from various fields of natural science and engineering, such as biology [1, 24, 39, 56], neuroscience [46], computer vision [17, 26, 36, 42, 62] and communication networks [27, 45, 58].

Theorem 1 in [64] asserts that the solution of the unlabeled sensing problem is indeed unique if $s \geq 2n$ and A^* is generic. For the case $m = s$, Song, Choi and Shi [58] proposed a new method for

recovering the vector ξ^* as follows: let

$$q_i(x) = p_i(A^*x) - p_i(y^*),$$

where $x = [x_1, \dots, x_n]$ and

$$p_k(y) = \sum_{i=1}^m y_i^k \in \mathbb{R}[y]$$

is the k -th power sum of the variables $y = [y_1, \dots, y_m]$. As p_k is a symmetric polynomial in y , its value is independent of the order of the variables y_1, \dots, y_m . Suppose π is an element of the permutation group Σ_m , if ξ^* is a solution of

$$A^*x = \pi(y^*), \quad (1)$$

then it is a root of the polynomials $q_i(x)$, i.e.,

$$q_i(\xi^*) = p_i(A^*\xi^*) - p_i(y^*) = p_i(\pi(y^*)) - p_i(y^*) = 0$$

for $i = 1, \dots, m$.

By Theorem 1 in [58] and by [64, 65], with $A^* \in \mathbb{R}^{m \times n}$ a given matrix with i.i.d random entries drawn from an arbitrary continuous probability distribution over \mathbb{R} , if $m \geq 2n$, then with probability 1, ξ^* is the unique solution of the polynomial system

$$Q_m = \{q_1(x) = 0, \dots, q_m(x) = 0\}. \quad (2)$$

Numerical experiments showed that solving the first $n+1$ equations is sufficient for recovering the solution ξ^* ; i.e. ξ^* is the unique solution of Q_{n+1} . Hence, [58] pointed out the following open problem (see also Conjecture 6 in [43]):

OPEN PROBLEM. For a generic matrix $A^* \in \mathbb{R}^{m \times n}$, $m \geq n+1$, and a permutation y^* of a generic vector in the column space of A^* , the $(n+1)$ -by- n polynomial system

$$Q_{n+1} = \{q_1(x) = 0, \dots, q_n(x) = 0, q_{n+1}(x) = 0\} \quad (3)$$

has a unique solution ξ^* , which satisfies $A^*\xi^* = \pi(y^*)$.

In [63], it is shown that for a generic matrix $A^* \in \mathbb{C}^{m \times n}$ and any vector y^* , the n -by- n polynomial system

$$Q_n = \{q_1(x) = 0, \dots, q_n(x) = 0\} \quad (4)$$

in the variables $x = x_1, \dots, x_n$ has at most $n!$ solutions. Furthermore, if y^* is a permutation of a generic vector in the column space of A^* , then among the solutions of Q_n there is only one vector ξ^* satisfying $A^*\xi^* = \pi(y^*)$. They can recover ξ^* by solving Q_n via symbolic or homotopy methods and then select ξ^* from $n!$ solutions via numerical optimization methods.



This work is licensed under a [Creative Commons Attribution International 4.0 License](https://creativecommons.org/licenses/by/4.0/).

ISSAC '24, July 16–19, 2024, Raleigh, NC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0696-7/24/07
<https://doi.org/10.1145/3666000.3669674>

EXAMPLE 1. Given a matrix A^* and a vector y^*

$$A^* := \begin{bmatrix} 1 & 2 \\ 4 & 3 \\ 0 & -2 \\ -2 & 0 \end{bmatrix}, y^* = \begin{bmatrix} -5 \\ -10 \\ 2 \\ 4 \end{bmatrix},$$

find a solution ξ^* such that

$$A^* \xi^* = \pi(y^*)$$

for an unknown permutation π of the coordinates of y^* .

From the matrix A^* and the vector y^* , we compute the polynomials

$$\begin{aligned} q_1(x) &= 3x_1 + 3x_2 + 9, \\ q_2(x) &= 21x_1^2 + 28x_1x_2 + 17x_2^2 - 145, \\ q_3(x) &= 57x_1^3 + 150x_1^2x_2 + 120x_1x_2^2 + 27x_2^3 + 1053, \\ q_4(x) &= 16x_1^3x_2 + 44x_1^2x_2^2 + 24x_1x_2^3 - 400. \end{aligned}$$

The polynomial system $\{q_1(x) = 0, q_2(x) = 0, q_3(x) = 0, q_4(x) = 0\}$ has a unique solution $\xi^* = (-1, -2)$. One can check that ξ^* is the solution of (1) for the permutation $\pi = [1, 2, 4, 3]$.

We can check that ξ^* is the unique solution of the polynomial system $\{q_1(x) = 0, q_2(x) = 0, q_3(x) = 0\}$, which gives an example supporting the positive answer to the open problem (1).

The polynomial system $\{q_1(x) = 0, q_2(x) = 0\}$ has two solutions $\eta^* = \left(-\frac{4}{5}, -\frac{11}{5}\right)$, $\xi^* = (-1, -2)$. One can check that η^* is not a solution of (1).

The following theorem shows that one can obtain the unique and desired vector ξ^* for a generic matrix A^* and a permutation y^* of a generic vector in the column space of A^* by solving the polynomial system Q_{n+1} .

THEOREM 1. The map

$$f : \mathbb{C}^{m \times n} \times \mathbb{C}^n \rightarrow \mathbb{C}^{m \times n} \times \mathbb{C}^{n+1} \quad (5)$$

$$(A^*, \xi^*) \mapsto (A^*, p_1(A^* \xi^*), \dots, p_{n+1}(A^* \xi^*))$$

is a birational morphism to the image $f(\mathbb{C}^{m \times n} \times \mathbb{C}^n) \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^{n+1}$.

REMARK 1. The fact that the morphism (5) is birational implies that there exists a dense open subset U_2 of V such that f induces a bijection from $U_1 := f^{-1}(U_2)$ to U_2 , see [22, Chapter I, Corollary 4.5]. Therefore, if the sample point (A^*, y^*) satisfies the condition $(A^*, p_1(y^*), \dots, p_{n+1}(y^*)) \in U_2$, then there exists a unique vector ξ^* such that $(A^*, \xi^*) \in U_1$ and

$$(A^*, p_1(A^* \xi^*), \dots, p_{n+1}(A^* \xi^*)) = (A^*, p_1(y^*), \dots, p_{n+1}(y^*)).$$

This answers the open problem above in the affirmative.

As an alternative to existing RANSAC [18], expectation maximization [2], branch & bound [53] or homotopy and Groebner bases [63] approaches for solving the unlabeled sensing problem in its generality (that is without special assumptions such as sparsity, as, e.g., considered in [57]), in this paper we focus on finding the unique solution of the polynomial system Q_{n+1} efficiently by reducing it to a rank-one moment matrix completion problem, which can then be solved by many efficient algorithms.

Structure of the paper. In Section 2, we briefly review the main idea in the proof of Theorem 1 in [38] and its relationship with previous known results. In Section 3, we show that solving polynomial system Q_{n+1} can be reduced to solving the rank-one moment matrix completion problem, which can be solved efficiently by solving a sequence of semidefinite programming (SDP) problems. In Section 4, we propose some symbolic and numerical algorithms based on Theorem 1 and 3. We conducted eight numeric and symbolic experiments to test the efficiency and robustness of the proposed algorithms.

2 PROOF OF THEOREM 1

For convenience, we will abbreviate the unlabeled sensing problem for case $m = s > n > 0$ by USP and call the data (A^*, y^*) in USP as the sample point of USP.

The morphism f can be written as the composition of the following two morphisms

$$g : \mathbb{C}^{m \times n} \times \mathbb{C}^n \rightarrow \overline{g(\mathbb{C}^{m \times n} \times \mathbb{C}^n)} \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m \quad (6)$$

$$(A^*, \xi^*) \mapsto (A^*, p_1(A^* \xi^*), \dots, p_m(A^* \xi^*))$$

$$\text{and } \gamma : \overline{g(\mathbb{C}^{m \times n} \times \mathbb{C}^n)} \rightarrow \overline{f(\mathbb{C}^{m \times n} \times \mathbb{C}^n)} \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m. \quad (7)$$

$$(A^*, p_1(\eta^*), \dots, p_m(\eta^*)) \mapsto (A^*, p_1(\eta^*), \dots, p_{n+1}(\eta^*)) \quad (8)$$

Since the composition of two birational morphisms is still a birational morphism, the proof of Theorem 1 can be divided into two parts:

- Show the morphism g is birational, and
- Show the morphism γ is birational.

When $m = n + 1$, the morphism g equals the morphism f in Theorem 1. We show first in Section 2.1 that the morphism g is birational. Hence, it implies Theorem 1 for the case $m = n + 1$, i.e., one can obtain the unique and desired vector ξ^* for a generic matrix A^* and a permutation y^* of a generic vector in the column space of A^* by solving the polynomial system Q_{n+1} for the case $m = n + 1$. Moreover, as g is birational, it implies that for generic sample points (A^*, y^*) of the unlabeled sensing problem (USP), the polynomial system Q_m in (2) has a unique solution $x = \xi^*$, which is also the unique solution of USP.

Since m can be much larger than n in the application, we need to show that f is birational for a general $m \geq n + 1$. This implies that the smaller polynomial system Q_{n+1} in (3) has a unique solution for generic (A^*, y^*) . The key point of showing that f is birational is to prove

$$[\mathbb{C}(A, x) : \mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))] = n!, \text{ and}$$

$$[\mathbb{C}(A, p_1(Ax), \dots, p_{n+1}(Ax)) : \mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))] = n!.$$

The first degree is the Bézout number of the complete intersection given by the regular sequence $p_1(Ax), \dots, p_n(Ax)$, and the second one is exactly the degree of the minimal polynomial of $p_{n+1}(Ax)$ over $\mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))$. In Section 2.2, we give a sketch proof showing that this polynomial is the resultant of the system $\{p_k(Ax) - z_k : k = 1, \dots, n + 1\}$.***

2.1 The morphism g is birational

In this subsection, we prove the following theorem, which implies that one can obtain the unique and desired vector ξ^* for a generic

matrix A^* and a permutation y^* of a generic vector in the column space of A^* by solving the polynomial system Q_m .

THEOREM 2. *For $m > n > 0$, the morphism*

$$g : \mathbb{C}^{m \times n} \times \mathbb{C}^n \rightarrow \mathbb{C}^{m \times n} \times \mathbb{C}^m$$

$$(A^*, \xi^*) \mapsto (A^*, p_1(A^* \xi^*), \dots, p_m(A^* \xi^*))$$

is birational onto the image $g(\mathbb{C}^{m \times n} \times \mathbb{C}^n)$.

When $m = n + 1$, the morphism g equals f , i.e., Theorem 2 implies Theorem 1 in this special case.

We decompose g into the composition of two dominant morphisms α and β defined by (9) and (12) respectively. To show g is birational, it suffices to show α and β are birational.

Define the morphism α

$$\alpha : \mathbb{C}^{m \times n} \times \mathbb{C}^n \rightarrow D_{n+1}(A|y) \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m, \quad (9)$$

$$(A^*, \xi^*) \mapsto (A^*, A^* \xi^*)$$

where $D_{n+1}(A|y)$ is the zero locus of the all $(n + 1)$ -minors of the matrix $(A|y)$, $A = [a_{ij}]$ is a m -by- n matrix of mn variables and $y = [y_i]$ is a column vector of m variables, $i \in [m]$, $j \in [n]$. Namely, $D_{n+1}(A|y)$ is the determinantal variety consisting of m -by- $(n + 1)$ matrices of rank less than $n + 1$. The determinantal variety $D_{n+1}(A|y)$ is irreducible [6, Proposition 1.1], i.e., it is not a union of any two proper closed subvarieties.

LEMMA 1. *The morphism α is birational.*

PROOF. The determinantal variety $D_{n+1}(A|y)$ is irreducible [6, (1.1) Proposition], and the morphism α maps the non-empty open subset S_0 of $\mathbb{C}^{m \times n} \times \mathbb{C}^n$ to the non-empty open subset $S_1 = \alpha(S_0)$ in $D_{n+1}(A|y)$, where

$$S_0 := \{(A^*, \xi^*) \in \mathbb{C}^{m \times n} \times \mathbb{C}^n : \text{rank}(A^*) = n\} \text{ and} \quad (10)$$

$$S_1 := \{(A^*, \eta^*) \in D_{n+1}(A|y) : \text{rank}(A^*) = n\} \quad (11)$$

are respectively the set of the solutions to USP and the set of correctly sorted sample points in USP. Hence, we deduce that α is dominant. Furthermore, for any $(A^*, \eta^*) \in S_1$, using Cramer's rule in linear algebra, we see that the fiber $\alpha^{-1}(A^*, \eta^*)$ is a singleton, hence the morphism α is restricted to a bijection from S_0 to S_1 . Therefore, α is birational. \square

The second dominant morphism, to be shown to be birational, is

$$\beta : D_{n+1}(A|y) \rightarrow \overline{\beta(D_{n+1}(A|y))} \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m \quad (12)$$

$$(A^*, \eta^*) \mapsto (A^*, p_1(\eta^*), \dots, p_m(\eta^*))$$

Note that β is the restriction of the finite morphism

$$\mathbb{C}^{m \times n} \times \mathbb{C}^m \rightarrow \mathbb{C}^{m \times n} \times \mathbb{C}^m, \quad (13)$$

$$(A^*, \eta^*) \mapsto (A^*, p_1(\eta^*), \dots, p_m(\eta^*))$$

hence β is also a finite and closed morphism [22, Chapter II, Exercise 3.5], and

$$\beta(D_{n+1}(A|y)) = \overline{\beta(D_{n+1}(A|y))} = \overline{g(\mathbb{C}^{m \times n} \times \mathbb{C}^n)}. \quad (14)$$

To prove the following Lemma 2, we shall first clarify the actions of Σ_m on both $\mathbb{C}^{m \times n} \times \mathbb{C}^m$ and $\mathbb{C}[A, y]$.

A permutation $\sigma \in \Sigma_m$ acts on $\mathbb{C}[A, y]$ as a \mathbb{C} -algebra isomorphism given by $\sigma(y_i) = y_{\sigma(i)}$ and fixing A . Then for any $f \in \mathbb{C}[A, y]$ and $y^* = (y_1^*, \dots, y_m^*) \in \mathbb{C}^m$, we have

$$(\sigma(f))(y^*) = f(y_{\sigma(1)}^*, \dots, y_{\sigma(m)}^*).$$

A permutation $\sigma \in \Sigma_m$ acts on $y^* = (y_1^*, \dots, y_m^*) \in \mathbb{C}^m$ by

$$\sigma(y_1^*, \dots, y_m^*) = (y_{\sigma^{-1}(1)}^*, \dots, y_{\sigma^{-1}(m)}^*). \quad (15)$$

Then $(\sigma(f))(y^*) = f(\sigma^{-1}(y^*))$. This action induces the fiberwise action of Σ_m on $\mathbb{C}^{m \times n} \times \mathbb{C}^m$, i.e., for $(A^*, y^*) \in \mathbb{C}^{m \times n} \times \mathbb{C}^m$, we define

$$\sigma(A^*, y^*) = (A^*, \sigma(y^*)). \quad (16)$$

For the variable vector $y = [y_1, \dots, y_m]$,

$$\sigma(y) := [y_{\sigma(1)}, \dots, y_{\sigma(m)}]$$

is also a vector of variables, and we denote by $D_{n+1}(A|\sigma(y))$ the determinantal variety on which all the $(n + 1)$ -minors of $(A|\sigma(y))$ vanish. Then we have

$$D_{n+1}(A|\sigma(y)) = \sigma(D_{n+1}(A|y)).$$

LEMMA 2. *The morphism β is birational.*

PROOF. To show that β is birational, it suffices to prove the following three facts

- The subset

$$W_1 := D_{n+1}(A|y) \setminus \bigcup_{\sigma \in \Sigma_m \setminus \{1\}} D_{n+1}(A|\sigma(y)) \quad (17)$$

is dense open in $D_{n+1}(A|y)$.

By definition, W_1 is an open subset of $D_{n+1}(A|y)$. According to [63, Proof of Theorem 1], for generic matrix $A^* \in \mathbb{C}^{m \times n}$, generic column vector $\xi^* \in \mathbb{C}^n$ and any m -permutation $\sigma \neq 1$, we have

$$\text{rank}(A^*|\sigma(A^* \xi^*)) = n + 1.$$

Hence W_1 is nonempty and dense open in $D_{n+1}(A|y)$.

- $\beta^{-1}(\beta(W_1)) = W_1$, and the restriction β' of β on W_1 is a bijection from W_1 to $\beta(W_1)$.

We first notice that for any (A^*, η^*) , all the entries in η^* are different from each other. Using Vieta's Theorem, we derive that for any $(A^*, \xi^*) \in \beta^{-1}(\beta(W_1))$, there is a permutation $\sigma \in \Sigma_m$ such that

$$\sigma(\xi^*) = \eta^*.$$

Thus $\sigma = 1$ and $(A^*, \xi^*) = (A^*, \eta^*) \in W_1$. Moreover, we also deduce that β' is injective.

It is clear that β' is surjective, so the bijectivity of β' follows.

- $\beta(W_1)$ is dense open in $\beta(D_{n+1}(A|y))$.

Vieta's Theorem implies that β is a finite morphism, hence β is a closed morphism. Since W_1 is open in $D_{n+1}(A|y)$ and $\beta^{-1}(\beta(W_1)) = W_1$, we conclude that $\beta(W_1)$ is also open in $\beta(D_{n+1}(A|y))$. \square

Now Theorem 2 follows, because g is the composition of the birational morphisms α and β .

REMARK 2. In this remark, we construct explicitly the open subset W of X such that if the sample point (A^*, y^*) of USP lies in W , then there exists a unique m -permutation π and a unique vector $\xi^* \in \mathbb{C}^n$ such that $A^* \xi^* = \pi(y^*)$.

- The set S of sample points (A^*, y^*) in USP is

$$S := \bigcup_{\sigma \in \Sigma_m} \sigma(S_1) \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m, \quad (18)$$

where S_1 is defined in (11), and the action of Σ_m on $\mathbb{C}^{m \times n} \times \mathbb{C}^m$ is defined by (15) and (16). S is a dense open subset of the Σ_m -equivariant closed subvariety

$$X := \bigcup_{\sigma \in \Sigma_m} \sigma(D_{n+1}(A|y)) \subseteq \mathbb{C}^{m \times n} \times \mathbb{C}^m, \quad (19)$$

A key observation is that, the finite morphism in (13) parameterizes the Σ_m -orbits of X .

- Since W_1 in (17) is nonempty, for any $\tau \in \Sigma_m$ the set

$$\tau(W_1) = \tau(D_{n+1}(A|y)) \setminus \bigcup_{\sigma \in \Sigma_m \setminus \{\tau\}} \sigma(D_{n+1}(A|y))$$

is also nonempty and dense open in $\tau(D_{n+1}(A|y))$. Therefore, the variety X has exactly $|\Sigma_m| = m!$ irreducible components $\sigma(D_{n+1}(A|y))$ for $\sigma \in \Sigma_m$.

- Write W_1 in (17), then the set

$$W := \bigcup_{\sigma \in \Sigma_m} \sigma(W_1),$$

is dense open in X , hence $S \cap W \neq \emptyset$. Since any point in W lies in a unique irreducible component of X , we conclude that if the sample point (A^*, y^*) of USP lies in W , then there exists a unique m -permutation π and a unique vector $\xi^* \in \mathbb{C}^n$ such that $A^* \xi^* = \pi(y^*)$. Therefore, the solution (A^*, ξ^*) to USP is unique for most sample points, and in this case (A^*, ξ^*) is exactly the solution of system Q_m in (2).

2.2 The morphism γ is birational

To show the morphism γ in (7) is birational, we need the projection

$$\delta : \overline{f(\mathbb{C}^{m \times n} \times \mathbb{C}^n)} \rightarrow \mathbb{C}^{m \times n} \times \mathbb{C}^n. \quad (20)$$

$$(A^*, p_1(\eta^*), \dots, p_{n+1}(\eta^*)) \mapsto (A^*, p_1(\eta^*), \dots, p_n(\eta^*)) \quad (21)$$

Notice that $(A^*, p_1(\eta^*), \dots, p_{n+1}(\eta^*))$ is a point in $f(\mathbb{C}^{m \times n} \times \mathbb{C}^n)$. Hence, δ is uniquely determined by (21).

Now we give the sketch of the proof that the morphism γ is birational.

- For generic $(A^*, w^*) \in \mathbb{C}^{m \times n} \times \mathbb{C}^n$, the fiber $(\delta\gamma)^{-1}(A^*, w^*)$ consists of at most $n!$ points [63, Theorem 2]. Since g is birational, we deduce that the fiber $(\delta\gamma)^{-1}(A^*, w^*)$ also generically consists of at most $n!$ points. Hence, it suffices to show that δ is also a generically finite morphism with generic covering degree $n!$.
- Let $A = [a_{ij}]_{m \times n}$, $t = [t_1, \dots, t_n]$ and $z = [z_1, \dots, z_{n+1}]$ be the matrix and vectors in variables a_{ij} , t_i , z_j respectively. Let $R(A, z) \in \mathbb{Z}[A, z]$ and $c(A) \in \mathbb{Z}[A]$ be the resultants of the polynomial systems

$$\{p_k(At) - z_k : k = 1, \dots, n+1\}, \quad (22)$$

$$\{p_k(At) : k = 1, \dots, n\} \quad (23)$$

eliminating the variables t_1, \dots, t_n , respectively. We shall show that

- (i) The leading term of $R(A, z)$ in z_{n+1} is $c(A)^{n+1} z_{n+1}^n$;
- (ii) $R(A, p_1(Ax), \dots, p_{n+1}(Ax)) = 0$;
- (iii) $R(A, z)$ is irreducible in $\mathbb{C}(A, z_1, \dots, z_n)[z_{n+1}]$.

– To show (i), we use Macaulay's assertions on resultants [41, Section 8] to deduce that $\deg_{z_{n+1}}(R(A, z)) \leq n!$ and the coefficient of z_{n+1}^n in $R(A, z)$ is $c(A)^{n+1}$. According to Lemma 4 in [63], $p_1(A^*t), \dots, p_n(A^*t)$ is a homogeneous regular sequence of $\mathbb{C}[t]$ for generic matrix $A^* \in \mathbb{C}^{m \times n}$, hence $p_1(At), \dots, p_n(At)$ is also a homogeneous $\mathbb{C}(A)[t]$ -regular sequence. Thus, the system (23) in variables t has no solution in the projective $(n-1)$ -space over the algebraic closure of $\mathbb{C}(A)$. Therefore, [41, Section 10] asserts that $c(A)$ is nonzero in $\mathbb{C}[A]$, whence the leading term of $R(A, z)$ in z_{n+1} is $c(A)^{n+1} z_{n+1}^n$. Moreover, the fact that $p_1(At), \dots, p_n(At)$ is a homogeneous $\mathbb{C}(A)[t]$ -regular sequence implies that the field extension degree $[\mathbb{C}(A, x) : \mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))] = n!$ is the product of the degrees of $p_k(At)$, $k \in [n]$ in t . This fact also implies that for generic $(A^*, w^*) \in \mathbb{C}^{m \times n} \times \mathbb{C}^n$, the fiber $(\delta\gamma)^{-1}(A^*, w^*)$ consists of $n!$ points.

– To show (ii), we notice that the system (22) substituting z_k with $p_k(Ax)$ has a solution $t = x$ in the affine space over the algebraic closure of $\mathbb{C}(A, p_1(Ax), \dots, p_{n+1}(Ax))$. According to arguments in [41, Section 10], we know that

$$R(A, p_1(Ax), \dots, p_{n+1}(Ax)) = 0.$$

- To show (iii), let \bar{A} be the first $n+1$ rows of A , and $A_{>n+1}$ be the last $m - (n+1)$ rows of A . Then
 - (a) $R(\bar{A}, z) \equiv R(A, z) \pmod{A_{>n+1}}$, and $c(\bar{A}) \neq 0$;
 - (b) The leading term of $R(\bar{A}, z)$ in z_{n+1} is $c(\bar{A})^{n+1} z_{n+1}^n$;
 - (c) $R(\bar{A}, p_1(\bar{A}x), \dots, p_{n+1}(\bar{A}x)) = 0$;
 - (d) $[\mathbb{C}(\bar{A}, x) : \mathbb{C}(\bar{A}, p_1(\bar{A}x), \dots, p_n(\bar{A}x))] = n!$.

Note that the birational morphism g for the case $m = n+1$ implies that $\mathbb{C}(\bar{A}, p_1(\bar{A}x), \dots, p_{n+1}(\bar{A}x)) = \mathbb{C}(\bar{A}, x)$, hence the polynomial $R(\bar{A}, p_1(\bar{A}x), \dots, p_n(\bar{A}x), z_{n+1})$ in z_{n+1} is irreducible over $\mathbb{C}(\bar{A}, p_1(\bar{A}x), \dots, p_n(\bar{A}x))$. Since $p_1(\bar{A}x), \dots, p_n(\bar{A}x)$ is $\mathbb{C}(\bar{A})[x]$ -regular, $R(\bar{A}, z)$ in z_{n+1} is also irreducible over $\mathbb{C}(\bar{A}, z_1, \dots, z_n)$. Using Gauss' Lemma, we see that any factorization $R(A, z) = uv$ in $\mathbb{C}(A, z_1, \dots, z_n)[z_{n+1}]$ induces a factorization $R(A, z) = u_1 v_1$ in $\mathbb{C}[A, z]$ satisfying

$$\deg_{z_{n+1}} u_1 = \deg_{z_{n+1}} u, \deg_{z_{n+1}} v_1 = \deg_{z_{n+1}} v$$

and the leading coefficients of u_1 and v_1 in z_{n+1} lie in $\mathbb{C}[A]$. Modulo $A_{>n+1}$ and noticing that $c(\bar{A})^{n+1} \neq 0$ is the leading coefficient of $R(\bar{A}, z)$, we deduce that $R(\bar{A}, z) = \bar{u}_1 \bar{v}_1$, $\deg_{z_{n+1}} \bar{u}_1 = \deg_{z_{n+1}} u_1$ and $\deg_{z_{n+1}} \bar{v}_1 = \deg_{z_{n+1}} v_1$. Thus, $R(A, z)$ is irreducible over $\mathbb{C}(A, z_1, \dots, z_n)$.

Hence, the algebraic field extension degree

$$[\mathbb{C}(A, p_1(Ax), \dots, p_{n+1}(Ax)) : \mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))] = n!$$

coincides with $[\mathbb{C}(A, x) : \mathbb{C}(A, p_1(Ax), \dots, p_n(Ax))] = n!$, which implies that δ is a generically finite morphism with a generic covering degree $n!$, and

$$\mathbb{C}(A, p_1(Ax), \dots, p_{n+1}(Ax)) = \mathbb{C}(A, x),$$

namely, γ is birational, and Theorem 1 follows. We refer to [38] for detailed proof of the above arguments.

REMARK 3. From the fact that γ is birational, we can also derive that for generic $(A^*, w^*) \in \mathbb{C}^{m \times n} \times \mathbb{C}^n$, the fiber $(\delta\gamma)^{-1}(A^*, w^*)$ exactly consists of $n!$ points, among which there is, however, only one solution to USP. This result indicates that the bound $n!$ in [63, Theorem 2] is optimal.

3 RANK-ONE MOMENT MATRIX COMPLETION

There are many well-known symbolic or numeric algorithms and software packages for solving zero-dimensional polynomial systems in variables $x = [x_1, \dots, x_n]$

$$\{g_1(x) = 0, \dots, g_m(x) = 0\}, \quad (24)$$

e.g. [3, 4, 19, 35, 55, 67]. A desirable method for us is the semidefinite relaxation method proposed by Parrilo [51, 52], Lasserre [33] for solving a sequence of SDP problems with constant objective function 1

$$\begin{cases} \min & 1 \\ \text{s. t.} & y_0 = 1, \\ & M_t(y) \geq 0, \\ & M_{t-d_j}(g_j y) = 0, \quad j = 1, \dots, m, \end{cases} \quad (25)$$

where $d_j := \lceil \deg(g_j)/2 \rceil$, $y = (y_\alpha)_{\alpha \in \mathbb{N}^n} \in \mathbb{R}^{\mathbb{N}^n}$, and $M_t(y)$ is the truncated moment matrix of order t

$$M_t(y) := (y_{\alpha+\beta})_{|\alpha|, |\beta| \leq t} \in \mathbb{R}^{\mathbb{N}^n \times \mathbb{N}^n},$$

$|\alpha| = \sum_i \alpha_i$, $|\beta| = \sum_i \beta_i$, which is the principal submatrix of the full (infinite) moment matrix

$$M(y) = (y_{\alpha+\beta}) \in \mathbb{R}^{\mathbb{N}^n \times \mathbb{N}^n}, \quad \alpha, \beta \in \mathbb{N}^n.$$

Suppose $g_j(x) = \sum_{\alpha \in \mathbb{N}^n} g_{j,\alpha} x^\alpha \in \mathbb{R}[x]$. If the (k, l) -th entry of $M_t(y)$ is y_β , then the (k, l) -th entry of the localizing matrix $M_{t-d_j}(g_j y)$ with respect to y and g_j is defined by

$$M_{t-d_j}(g_j y)(k, l) := \sum_{\alpha} g_{j,\alpha} y_{\alpha+\beta}.$$

Lasserre, Laurent, and Rostalski gave explicitly rank conditions that guarantee us to find all real solutions of the polynomial system (24) [30–32] by the semidefinite relaxation method. Let $d = \max_{j=1, \dots, m} d_j$, for $t \geq d$, define the convex set

$$K_t^{\mathbb{R}} := \left\{ y \in \mathbb{R}^{\mathbb{N}^n} \mid y_0 = 1, M_t(y) \geq 0, M_{t-d_j}(g_j y) = 0, j = 1, \dots, m \right\} \quad (26)$$

PROPOSITION 1. [30, Proposition 4.4] Let $t \geq d$ and $y \in K_t^{\mathbb{R}}$ for which $\text{rank} M_t(y)$ is maximum. If

$$\text{rank} M_s(y) = \text{rank} M_{s-1}(y) \quad (27)$$

holds for some $2d \leq s \leq t$, or

$$\text{rank} M_s(y) = \text{rank} M_{s-d}(y) \quad (28)$$

for some $d \leq s \leq t$. Then $I(V_{\mathbb{R}}(I)) = \langle \text{Ker} M_s(y) \rangle$, where $I = \langle g_1, \dots, g_m \rangle$. Moreover, $\text{rank} M_s(y) = |V_{\mathbb{R}}(I)|$.

The following result is obtained by applying Proposition 1 to the polynomial system with a unique real solution.

THEOREM 3. If the polynomial system $\{g_1(x) = 0, \dots, g_m(x) = 0\}$ has a unique real solution, suppose $y \in K_d^{\mathbb{R}}$ satisfies the flat extension condition (28), then the unique real solution can be read from the first column of the truncated moment matrix $M_d(y)$ for $d = \max_{j=1, \dots, m} \lceil \deg(g_j)/2 \rceil$.

PROOF. Since the flat extension condition (28) is satisfied, Corollary 1.4 in [34] shows that the vector $y \in K_d^{\mathbb{R}}$ can be extended to $\tilde{y} \in K^{\mathbb{R}} := \{y \in \mathbb{R}^{\mathbb{N}^n} \mid y_0 = 1, M(y) \geq 0, M(g_j y) = 0, j = 1, \dots, m\}$, with $\text{rank}(M(\tilde{y})) = \text{rank}(M_d(y)) = \text{rank}(M_0(y)) = 1$. When the polynomial system has a unique solution, $\text{rank}(M(\tilde{y})) = 1 = |V_{\mathbb{R}}(I)|$, according to Proposition 1.1 in [30], $I(V_{\mathbb{R}}(I)) = \ker(M(\tilde{y}))$. Moreover, Proposition 3.6 in [30] shows $\ker(M(\tilde{y})) = \langle \ker(M_d(y)) \rangle$. Now Theorem 3.3 and Proposition 1.1 in [30] imply the unique solution ξ^* can be recovered from the first column of $M_d(y)$, whose $(1, 1)$ entry is 1, and $\xi_i^* = M(\tilde{y})(1, i+1) = M_d(y)(1, i+1)$, $i = 2, \dots, n$. \square

REMARK 4. For solving a zero-dimensional polynomial, it is well-known that the flatness conditions in Proposition 1 will be satisfied for some $s \geq d$. In Theorem 3, $y \in K_d^{\mathbb{R}}$ satisfying condition (28) exists and is unique, which is exactly the moment sequence generated by the Dirac measure δ_{ξ^*} , to be more concrete $y_\alpha = (\xi^*)^\alpha$. When condition (28) is achieved at $t = s = d$, it is sufficient for recovering the solution from $M_d(y)$, and the size of moment matrices for SDP solvers is therefore bounded by $\binom{n+d}{d}$.

According to Theorem 3, to obtain the unique solution of I , one must find the truncated moment sequence $y \in K_d^{\mathbb{R}}$ with rank 1. Hence, finding the unique real solution of a polynomial system is reduced to solving the rank-one moment matrix completion problem. There are many well-known methods for solving low-rank matrix completion problems [10–12, 14–16, 21, 23, 25, 48–50, 54, 66].

The nuclear norm of a matrix is the sum of the singular values of the matrix. If the matrix is symmetric and positive semidefinite, then its nuclear norm is equal to the trace of the matrix. In [54, Theorem 2.2], Recht, Fazel, and Parrilo have shown that the nuclear norm is the best convex lower approximation of the rank function over the set of matrices with spectral norm less than or equal to 1. Therefore, the problem of finding a rank-one moment matrix can be relaxed to the following form:

$$\begin{cases} \min & \text{tr}(M_t(y)) \\ \text{s. t.} & y_0 = 1, \\ & M_t(y) \geq 0, \\ & M_{t-d_j}(g_j y) = 0, \quad j = 1, \dots, m \end{cases} \quad (29)$$

In [14, 15], Cossic and Demanet also show that when the rank-one matrix completion problem has a unique solution, the second-order semidefinite relaxation with minimization of the nuclear norm will be enough for finding the unique solution. In [40, 68, 69], we have shown that by solving the nuclear norm optimization problem (29),

one can find some real solutions (not all solutions) for polynomial systems more efficiently.

According to Theorem 1, the polynomial system Q_{n+1} has a unique real solution. Hence, in the next section, instead of minimizing the constant objective function 1 in (25), we solve the nuclear norm minimization problem (29) to find the unique real solution of Q_{n+1} .

4 EXPERIMENTS

4.1 Algorithm Design

For simplicity, we assume that our data $A^* \in \mathbb{R}^{m \times n}$, $y^* \in \mathbb{R}^n$ are generic in the sense of Theorem 1 and [63, Theorem 2]. This section presents two symbolic algorithms and one symbolic-numeric algorithm to solve the unlabeled sensing problem (1).

A straightforward attempt to recover ξ^* is to solve the polynomial system Q_n using Groebner basis [7–9], which is formalized as the following algorithm. The experiment 4.2.1 was conducted on Maple's symbolic computation software, and Maple automatically chose the monomial order, usually the reverse-graded lexicographic order.

Algorithm 1: Groebner basis for solving square system Q_n (4)

Input: $A^* \in \mathbb{Q}^{m \times n}$, $y^* \in \mathbb{Q}^n$ such that $\exists \pi \in \Sigma_m$, the permutation group, and $\pi(y^*) \in \text{im}(A^*)$, the image space of A^*
Output: ξ^* such that $A^* \xi^* = \pi(y^*)$
 compute Groebner basis of the ideal $I_n = (Q_n)$, and extracted the roots symbolically
 $roots \leftarrow \{\xi \in \mathbb{Q}^n \mid q_i(\xi) = 0, i = 1, \dots, n\};$
for $\xi \in roots$ **do**
 if *coordinates of $A^* \xi$ are a permutation of y^** **then**
 return $\xi^* \leftarrow \xi$
end
end

The complexity of solving zero-dimensional polynomial systems in n variables is known to be single exponential in n [13, 19, 28], even for well-behaved cases [20]. The complexity of examining whether coordinates of $A^* \xi$ are a permutation of y^* is quasi-linear in m using a quick sorting algorithm.

According to Theorem 1, the polynomial system Q_{n+1} has only one solution. Experimental results in Section 4.2 show that it is far more efficient to compute the Groebner basis for polynomial systems with only one solution. This observation leads to the following algorithm 2.

Both algorithms above use symbolic computation. Given precise input and clean measurement, the solution one obtains using these two algorithms is exact. When the given data are corrupted by noise, the perturbed polynomial system Q_{n+1} will have no solution as it is an overdetermined polynomial system. However, in real-world applications, all measurements are inevitably influenced by noise. Therefore, a robust and efficient symbolic-numerical algorithm is needed to deal with corrupted measurements y^* one observes in the unlabeled sensing problem. In Section 3, we have shown that

Algorithm 2: Groebner basis for solving overdetermined system Q_{n+1} (3)

Input: $A^* \in \mathbb{Q}^{m \times n}$, $y^* \in \mathbb{Q}^n$ such that $\exists \pi \in \Sigma_m$, the permutation group, and $\pi(y^*) \in \text{im}(A^*)$, the image space of A^*
Output: ξ^* such that $A^* \xi^* = \pi(y^*)$
 compute Groebner basis of the ideal $I_{n+1} = (Q_{n+1})$, and extracted the roots symbolically
 $roots \leftarrow \{\xi \in \mathbb{Q}^n \mid q_i(\xi) = 0, i = 1, \dots, n+1\};$
 $roots = \{\xi^*\};$
return ξ^*

the unique solution of the polynomial system Q_{n+1} can be obtained by solving the rank-one moment matrix completion problem (29). Below, we develop the rank-one moment matrix completion method for solving polynomial system Q_{n+1} (3) numerically.

Algorithm 3: Matrix completion solving overdetermined system Q_{n+1} (3)

Input: $A^* \in \mathbb{R}^{m \times n}$, $y^* \in \mathbb{R}^n$ such that $\exists \pi \in \Sigma_m$, $\pi(y^*) \in \text{im}(A^*)$
Output: ξ^* such that $A^* \xi^* = \pi(y^*)$
 solve SDP problem (29) for $g_i = q_i$ to the relaxation order $s = \lceil (n+1)/2 \rceil$ to get the moment matrix M_s
 compute singular value decomposition of M_s
 $[U, S, V] \leftarrow \text{svd}(M_s);$
 select the first $n+1$ rows of U
 $x_1 \leftarrow U(1 : n+1, 1);$
 normalize x_1 to recover ξ_{sdp}
 $\xi_{sdp} \leftarrow (\xi(k) \leftarrow x_1(k+1, 1)/x_1(1, 1))_{k=1, \dots, n};$
 use EM method in [61] to refine ξ_{sdp}
 $\xi_{EM} \leftarrow EM(\xi_{sdp});$
return $\xi^* \leftarrow \xi_{EM}$

Semidefinite programming problems can be solved using the interior-point method [5], whose complexity is bounded by a polynomial in the matrix size. There are well-developed software packages to solve SDP problems, for example, Sedumi [59] and SDPNAL+ [60].

The Expectation-Maximization (EM) method proposed in [63] is an algebraic trick to refine the solution extracted from a numerical polynomial system solver. By sorting $A^* \xi_{sdp}$ and y^* , if the solution extracted is accurate enough, one expects that $A^* \xi_{sdp}$ and y^* after sorting should match, hence recovers the permutation π and reduces the problem to classical linear regression. The numerical precision of classical linear regression is determined by the condition number of A^* and precision of y^* , which, in practice, can be better controlled in numerical algorithms.

4.2 Symbolic and Numerical Experimental Results

The experiments were conducted on a desktop with an Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz and 128 GB RAM. The matrix entries of A^* were randomly generated from standard Gaussian

distribution in these experiments, and entries ξ^* were generated from a uniform distribution in $(0, 1)$. Equations $(q_i = 0)_{i=1,\dots,n+1}$ were then constructed from A^* , ξ^* , and possible noise. SDP in algorithm 3 was solved using Sedumi [59]. Source codes are uploaded on GitHub at [unlabeled-sensing-ISSAC-2024](https://github.com/unlabeled-sensing-ISSAC-2024).

4.2.1 symbolic algorithm 1 and 2, clean input data in \mathbb{Q} , algorithm performance on different n .

Experiment 4.2.1 compares the running time of algorithm 1 (TA1/s) and algorithm 2 (TA2/s) with different $n = 3, 4, \dots, 8$, and m was set as $2n$, clean measurement. Algorithm 1 uses the square system $Q_n = \{q_1(x) = 0, \dots, q_n(x) = 0\}$. Algorithm 2 uses the overdetermined system $Q_{n+1} = \{q_1(x) = 0, \dots, q_n(x), q_{n+1}(x) = 0\}$. All of the Groebner basis computations are done in Maple.

Table 1: cpu time for Algorithm 1 and Algorithm 2, with different $n, m = 2n$.

n	TA1/s	TA2/s
3	0.05	0.08
4	1.95	0.24
5	153.05	0.56
6	–	3.30
7	–	200.39
8	–	72549.63

Table 1 shows that as the number of variables and the degree of the polynomial system increase, computing a Groebner basis for the polynomial system Q_n or Q_{n+1} becomes more difficult. As Q_{n+1} has only one solution, its Groebner basis in lexicographic order consists of n linear polynomials $x_1 - \xi_1^*, \dots, x_n - \xi_n^*$. On the other hand, the polynomial system Q_n has at most $n!$ solutions; its Groebner basis consists of a set of polynomials with higher degrees and larger coefficients. Therefore, solving the overdetermined polynomial system Q_{n+1} is much faster than solving the square polynomial system Q_n . When $n \geq 6$, algorithm 1 could not return the result within 24 hours and was terminated manually.

4.2.2 numerical algorithm 3, clean input data in \mathbb{R} , algorithm performance on different n .

The following experiments examine the performance of numerical algorithm 3. Since algorithm 3 computes the moment optimization problem numerically, the relative error of the solution extracted from the moment matrix is defined as $err_{sdp} := \frac{\|\xi^* - \xi_{sdp}\|_2}{\|\xi^*\|_2}$, where ξ_{sdp} represents the solution extracted by semidefinite programming part of algorithm 3.

Experiment 4.2.2 examines the running time of algorithm 3 (TA3/s) with different $n = 3, 4, 5, 6$, and m was set as $2n$, t records the relaxation order used in the semidefinite programming, $ranks$ records the rank sequence of the truncated moment matrix M , $size$ denotes the dimension of the truncated moment matrix for the given relaxation order. In this experiment, no noise was imposed on y .

Table 2 shows that as the number of variables and the degree of polynomial rise, the time consumed to compute the moment matrix increases. The rank sequence of the moment matrix M stabilizing at 1 indicates that the semidefinite programming reaches the flat extension criterion at the lowest relaxation order required, which

Table 2: CPU time, relative error of ξ_{sdp} , rank sequence and moment matrix size for Algorithm 3 with different $n, m = 2n$.

n	t	TA3/s	err_{sdp}	$ranks$	$size$
3	2	0.20	2.73E-08	1, 1, 1	10
4	3	1.88	6.53E-07	1, 1, 1, 1	35
5	3	5.58	1.62E-06	1, 1, 1, 1	56
6	4	290.95	4.93E-06	1, 1, 1, 1, 1	210

is $\lceil (n+1)/2 \rceil$, therefore Theorem 3 guarantees that the solution extracted from the M is exactly the unique solution of the polynomial system Q_{n+1} . When using clean measurement y , the err_{sdp} behaves well and mildly increases as n becomes larger for $n = 3, 4, 5, 6$.

4.2.3 numerical algorithm 3, corrupted measurement y , algorithm performance on different m .

The following experiments included noise on measurement. For real solution ξ^* , and clean measurement $y^* := A^* \xi^*$, noise y_c was assumed to be a Gaussian random vector with expectation 0 and covariance matrix $\sigma^2 I$, where I denotes the $n \times n$ identity matrix. The magnitude of noise in signal processing is measured using signal-noise-ratio SNR in decibels. In our experiments, SNR can be computed from the formula

$$SNR = 10 \left(\log_{10} \left(\frac{n}{3\sigma^2} \right) \right).$$

The equations generated using corrupted measurement $y := y^* + y_c$ as

$$q_{ic} := p_i(A^* x) - p_i(y) = 0.$$

We further report the relative error of the solution refined by EM algorithm [63] as $err_{EM} := \frac{\|\xi^* - \xi_{EM}\|_2}{\|\xi^*\|_2}$. EM method is essentially a sorting procedure. It sorts y according to its coordinates to y_{sort} . For possible solutions $(x_i)_i$, compute Ax_i and sort its coordinations to $(Ax_i)_{sort}$. The most likely solution should be the nearest one to y_{sort} . Hence, the permutation is determined accordingly. Once the permutation is found, solving USP is reduced to solving a linear system.

Experiment 4.2.3 examines the running time and relative error of algorithm 3 with respect to m under the noise of SNR = 60dB, fixing $n = 4$. T_{EM} , T_{sdp} record the CPU time of EM refinement and solving SDP problem in algorithm 3, $TA3 = T_{EM} + T_{sdp}$ is the total CPU time of algorithm 3.

Table 3: CPU time, and relative error of ξ_{EM} and ξ_{sdp} for algorithm 3 with different m , observation y is corrupted by Gaussian random noise of SNR = 60dB, $n = 4$ fixed, 20 trials median.

m	err_{EM}	err_{sdp}	T_{EM}/s	T_{sdp}/s	TA3/s
500	0.010%	0.141%	0.003	2.477	2.450
1000	0.006%	0.102%	0.003	2.586	2.589
2000	0.005%	0.125%	0.016	2.602	2.617
5000	0.006%	0.250%	0.188	2.672	2.859

REMARK 5. Algorithm 3 may fail to find a good approximate solution using SDP solvers and EM refinement possibly cannot recover the right permutation from the solution extracted from ξ_{sdp} , leading to the relative error of the solution returned by algorithm 3 diverging

very far away from most of the trials. Here we manifest a typical outlier on the condition $n = 4, m = 500, \text{SNR} = 60\text{dB}$, the median of relative errors of the trials are $\text{err}_{EM} = 0.016\%$ and $\text{err}_{sdp} = 0.396\%$, the outlier appeared in our experiment is $\text{err}_{EM} = 165.957\%$ and $\text{err}_{sdp} = 167.938\%$. For small $n = 4$ and noise $\text{SNR} = 80\text{dB}$ or $\text{SNR} = 100\text{dB}$, the proportion of the outliers is less than 5%. The proportion of the outliers is between 5% and 35% for other n and SNR , increasing along with the number of variables and noise.

The outliers are mainly caused by noise interfering SDP algorithm. The rank sequence of SDP relaxation without noise under the assumption of a unique solution should stabilize at 1. However, in outliers, the numerical rank sequence increases beyond 1 under a certain numerical tolerance concerning the noise. Set $n = 4, m = 50, \text{SNR} = 60\text{dB}$ and tolerance 0.01 for the numerical rank. In 50 trials, 2 outliers were recorded; these two outliers have numerical rank sequence 1, 3, 5, 5, and 1, 2, 3, 3. Many factors may lead to the increase of the rank sequence. We observed that the scale of the coordinates of the real solution ξ^* varied acutely in two outliers. The outlier with numerical rank sequence 1, 3, 5, 5 corresponds to the real solution $\xi^* = (0.753, 0.081, 0.326, 0.879)^t$; the outlier with rank sequence 1, 2, 3, 4 corresponds to the real solution $\xi^* = (0.414, 0.515, 0.004, 0.624)^t$.

Table 3 shows that the accuracy err_{sdp} and err_{EM} of algorithm 3 is stable with different m . As m increases, the time $TA3$ consumed mildly rises because the EM refinement part of algorithm 3 needs to sort more entries. However, compared with the sorting procedure, the main computation in algorithm 3 occurs in solving the SDP problem, which is irrelevant to m . Therefore, $TA3$ increases rather slowly when m becomes larger. Table 3 also reveals that for $m = 500$ or 1000 , the portion of time consumed on EM refinement is insignificant in the overall time consumed; hence, the overall time of algorithm 3 will be reported in further experiments, and the time of EM refinement will be omitted in tables below.

4.2.4 numerical algorithm 3, corrupted measurement y , algorithm performance on different magnitude of noise SNR , $n = 5, 6$.

Experiment 4.2.4 examines for each n the running time and relative error of algorithm 3 with respect to the magnitude of noise, fixing $m = 500$.

Table 4: CPU time, and relative error of ξ_{EM} and ξ_{sdp} for algorithm 3 with different magnitude of noise, $n = 5, m = 500$ fixed, 20 trials median.

SNR/dB	TA3/s	err_{sdp}	err_{EM}
100	5.46	0.003%	0.003%
80	5.58	0.043%	0.021%
60	6.50	0.220%	0.010%
50	6.97	1.430%	0.058%

Tables 4, 5 show that as the noise builds up, the err_{sdp} becomes larger, and EM refinement technique successfully improves the precision of the solution in all these experiments. For small noise $\text{SNR} = 100\text{dB}$ or 80dB , the err_{EM} is not sensitive to n . This is not so strange because the precision of ξ_{EM} depends majorly on the sorting procedure. When the noise is relatively small, with high probability EM technique finds the right permutation π . Once the permutation

Table 5: CPU time, and relative error of ξ_{EM} and ξ_{sdp} for algorithm 3 with different magnitude of noise, $n = 6, m = 500$ fixed, 20 trials median.

SNR/dB	TA3/s	err_{sdp}	err_{EM}
100	185.83	0.002%	0.002%
80	184.59	0.023%	0.019%
60	253.38	0.546%	0.011%
50	220.52	1.224%	0.044%

π is deduced correctly, the precision of classical linear regression only depends on the condition number of A and the relative noise $\|\frac{y_c}{A^* \xi^*}\|_2$ imposed on the measurement. For general $n = 5, 6$, the condition number of A with random entries of standard normal distribution behaves well, and the magnitude of relative noise is less sensitive to n . Hence, the relative error after EM refinement err_{EM} behaves well in small noise circumstances.

4.2.5 comparison between algorithm AIEM [63] and algorithm 3.

Algorithm AIEM was introduced in [63]. AIEM uses homotopy method [44] to solve the polynomial system Q_n (4). Homotopy continuation method solves polynomial systems by transforming the original system into a continuous path from an easily solvable system to the desired one. It introduces an auxiliary parameter t and constructs a family of homotopy equations with t , which smoothly deforms from a known system to the target system to be solved. The homotopy solver returns at most $n!$ roots, among which the real solution will be selected by EM sorting procedure. The relative error for ξ_{homo} the solution from homotopy solver after selection by sorting is defined as: $\text{err}_{homo} := \frac{\|\xi^* - \xi_{homo}\|_2}{\|\xi^*\|_2}$.

Table 6: CPU time and relative error for AIEM and algorithm 3 with different n . $m = 500, \text{SNR} = 80\text{dB}$, 20 trials median.

n	AIEM(homotopy)			algorithm 3(SDP)		
	TA_{AIEM}/s	err_{homo}	err_{EM}	$TA3/s$	err_{sdp}	err_{EM}
3	0.12	0.043%	0.012%	0.29	0.029%	0.016%
4	0.23	0.037%	0.016%	3.04	0.021%	0.019%
5	0.79	0.040%	0.014%	5.58	0.043%	0.021%
6	9.06	0.039%	0.019%	184.80	0.053%	0.011%

Table 6 shows that as n increases, AIEM using the homotopy solver HOM4PS2 [37] is more efficient than algorithm 3 using the SDP solver Sedumi. The SDP solver Sedumi used in algorithm 3 implemented the traditional interior-point method in its inner iterations, which is numerically more stable but is relatively slow compared to modern SDP solvers such as SDPNAL+. In [63], the homotopy solver Bertini [3] was used to solve the polynomial system Q_n , for $n = 6$ AIEM needs 2243 seconds to return the solution. Our experiments suggest that different software and implementations can influence the time and accuracy of AIEM and algorithm 3, which should be further examined and compared in the future.

4.2.6 Computational complexity analysis.

The complexity of solving a zero-dimensional polynomial system using Groebner bases is single exponential $d^{O(n)}$ [29], where d is

the degree of polynomials and n is the number of variables. In the context of unlabeled sensing and algorithm 1 ($d = n$), $2(d = n + 1)$, the overall complexity of symbolic algorithm should be $n^{O(n)} = O(\exp(n \log n))$.

For general input, $\exists n!$ solutions for homotopy continuation algorithm, and EM procedure picks one among the $n!$ solutions [63]. Fixing the precision of the result, the complexity of the homotopy continuation algorithm is proportional to the number of solutions, providing a lower bound $\Omega(n!)$.

For the SDP relaxation algorithm 3, the interior method costs $O(N\sqrt{M})$, where N is the cost of solving Newton system in iterations, M characterizes the scale of semi-definite constraints [47]. Denote m_s as the size of the moment matrix. In the context of unlabeled sensing and algorithm 3, $N = O(m_s^6)$, $M = O(m_s n)$, and $m_s \leq \binom{n+d}{d} = \binom{2n+1}{n}$. Fixing the precision of results, an upper bound of the complexity of algorithm 3 can be estimated as $O\left(\binom{2n+1}{n}^{6.5} \sqrt{n}\right)$. Comparing the complexity of homotopy continuation algorithm and algorithm 3:

$$\frac{\text{cost}(HC)}{\text{cost}(SDP)} = \Omega\left(\frac{n!}{\binom{2n+1}{n}^{6.5} \sqrt{n}}\right) \subseteq \Omega(\exp(n)), \quad (30)$$

we know that in the perspective of computational complexity, algorithm 3 has an exponential speed-up over the homotopy continuation algorithm.

For the case $n = 5$, the homotopy continuation algorithm in Bertini took half an hour to find 120 solutions [63], while the SDP algorithm in Sedumi used only 5.58 seconds to find the unique solution. In our experiment, we used HOM4PS2 to solve the polynomial system, and it took only 0.79 seconds to find 120 solutions. Different implementations seem to influence the actual runtime of the algorithms. For low-rank matrix completion problems, there are also other practical algorithms; future research should test other implementations.

5 CONCLUSION

In this paper, we give a positive answer to the open problem 1 by showing the polynomial system Q_{n+1} consisting of the $n+1$ Newton polynomials has a unique solution. The main Theorem 1 is proved by considering the birationality of the algebraic morphisms defined among several algebraic varieties. Since Q_{n+1} has a unique solution, the unlabeled sensing problem can be reduced to the classic rank-one moment matrix completion problem, and the unique solution can be recovered at the lowest relaxation order $\lceil \frac{n+1}{2} \rceil$.

Enlightened by the main Theorem 1 and 3, we develop symbolic and numerical algorithms 1, 2 and 3. Algorithm 1 and 2 are based on the Groebner basis computation, and algorithm 3 is based on semidefinite programming.

We test and compare the performance of these algorithms for different n, m and SNR . Our experiments show that for clean input without noise, algorithm 2 solving the overdetermined system Q_{n+1} is faster than algorithm 1, 3 and the existing algorithm AIEM, as the Groebner basis of the ideal generated by polynomials in Q_{n+1} has simple linear form for lexicographic monomial order and graded monomial order. For corrupted input data with noise, our experiment shows that algorithm 3 can return an approximate solution

within several minutes. Besides, our experiments confirm the results in Theorem 3. The rank sequence of the moment matrices in algorithm 3 stabilizes at 1 within the numeric tolerance when using clean input data. However, we also notice that due to the degree of the polynomial system Q_{n+1} is relatively high for the moment relaxation method, algorithm 3 may fail to recover the unique solution, which should be improved in the future.

ACKNOWLEDGMENTS

Lihong Zhi is supported by the National Key R&D Program of China (2023YFA1009401) and the National Natural Science Foundation of China (12071467). Manolis C. Tsakiris is supported by the National Key R&D Program of China (2023YFA1009402). Jingyu Lu would like to thank the help of Tianshi Yu on the experiments. The authors acknowledge the support of the Institut Henri Poincaré (UAR 839 CNRS-Sorbonne Université) and LabEx CARMIN (ANR-10-LABX-59-01).

REFERENCES

- [1] Abubakar Abid and James Zou. 2018. A Stochastic Expectation-Maximization Approach to Shuffled Linear Regression. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 470–477. <https://doi.org/10.1109/ALLERTON.2018.8635907>
- [2] A. Abid and J. Zou. 2018. A Stochastic Expectation-Maximization Approach to Shuffled Linear Regression. In *Annual Allerton Conference on Communication, Control, and Computing*. 470–477.
- [3] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. 2019. Bertini: Software for Numerical Algebraic Geometry. (2019). <https://doi.org/>
- [4] J  r  my Berthomieu, Christian Eder, and Mohab Safey El Din. 2021. Msolve: A library for solving polynomial systems. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*. 51–58.
- [5] Stephen P Boyd and Lieven Vandenbergh. 2004. *Convex optimization*. Cambridge university press.
- [6] Winfried Bruns and Udo Vetter. 1988. *Determinantal Rings*. Lecture Notes in Mathematics, Vol. 1327. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/BFb0080378>
- [7] Bruno Buchberger. 1965. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Innsbruck.
- [8] Bruno Buchberger. 1970. An Algorithmic Criterion for the Solvability of an Algebraic System of Equations. *Aequationes Mathematicae* 4 (1970), 374–383.
- [9] B. Buchberger. 1979. A criterion for detecting unnecessary reductions in the construction of Gr  bner-bases. In *Symbolic and Algebraic Computation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3–21.
- [10] Emmanuel Candes and Benjamin Recht. 2012. Exact matrix completion via convex optimization. *Commun. ACM* 55, 6 (2012), 111–119.
- [11] Emmanuel J. Candes and Yaniv Plan. 2010. Matrix Completion With Noise. *Proc. IEEE* 98, 6 (2010), 925–936. <https://doi.org/10.1109/JPROC.2009.2035722>
- [12] Emmanuel J. Candes and Benjamin Recht. 2008. Exact low-rank matrix completion via convex optimization. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. 806–812. <https://doi.org/10.1109/ALLERTON.2008.4797640>
- [13] Leandro Caniglia, Andr   Galligo, and Joos Heintz. 1988. Some new effectivity bounds in computational geometry. In *International Conference on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*. Springer, 131–151.
- [14] Augustin Cosse and Laurent Demanet. 2015. Rank-one matrix completion is solved by the sum-of-squares relaxation of order two. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. 9–12. <https://doi.org/10.1109/CAMSAP.2015.7383723>
- [15] Augustin Cosse and Laurent Demanet. 2021. Stable rank-one matrix completion is solved by the level 2 Lasserre relaxation. *Foundations of Computational Mathematics* 21 (2021), 891–940.
- [16] Alex P da Silva, Pierre Comon, and Andr   LF de Almeida. 2016. A finite algorithm to compute rank-1 tensor approximations. *IEEE Signal Processing Letters* 23, 7 (2016), 959–963.
- [17] Philip David, Daniel DeMenthon, Ramani Duraiswami, and Hanan Samet. 2004. SoftPOSIT: Simultaneous Pose and Correspondence Determination. *International Journal of Computer Vision* 59 (2004), 259–284. Issue 3. <https://doi.org/10.1023/B:VISI.0000025800.10423.1f>

- [18] G. Elhami, A. Scholefield, B. Béjar Haro, and M. Vetterli. 2017. Unlabeled sensing: Reconstruction algorithm and theoretical guarantees. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 4566–4570.
- [19] Jean-Charles Faugère. 1999. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra* 139, 1 (1999), 61–88. [https://doi.org/10.1016/S0022-4049\(99\)00005-5](https://doi.org/10.1016/S0022-4049(99)00005-5)
- [20] Jean-Charles Faugère, Mohab Safey El Din, and Thibaut Verron. 2016. On the complexity of computing Gröbner bases for weighted homogeneous systems. *Journal of Symbolic Computation* 76 (2016), 107–141.
- [21] M. Fazel, E. Candes, B. Recht, and P. Parrilo. 2008. Compressed sensing and robust recovery of low rank matrices. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*. 1043–1047. <https://doi.org/10.1109/ACSSC.2008.5074571>
- [22] Robin Hartshorne. 1977. *Algebraic Geometry* (1 ed.). Graduate Texts in Mathematics, Vol. 52. Springer, New York, NY. <https://doi.org/10.1007/978-1-4757-3849-0>
- [23] Didier Henrion, Simone Naldi, and Mohab Safey El Din. 2020. Real root finding for low rank linear matrices. *Applicable Algebra in Engineering, Communication and Computing* 31, 2 (2020), 101–133.
- [24] Xiaohu Huang and Anup Madan. 1999. CAP3: A DNA Sequence Assembly Program. *Genome Research* 9, 9 (1999), 868–877. <https://doi.org/10.1101/gr.9.9.868>
- [25] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 665–674.
- [26] Ruiping Ji, Yan Liang, Linfeng Xu, and Wanying Zhang. 2019. A Concave Optimization-Based Approach for Joint Multi-Target Track Initialization. *IEEE Access* 7 (2019), 108551–108560. <https://doi.org/10.1109/ACCESS.2019.2933650>
- [27] L. Keller, M. Jafari Siavoshani, C. Fragouli, K. Argyraki, and S. Diggavi. 2009. Identity Aware Sensor Networks. In *IEEE INFOCOM 2009*. 2177–2185. <https://doi.org/10.1109/INFCOM.2009.5062142>
- [28] Yagati N Lakshman. 1991. A single exponential bound on the complexity of computing Gröbner bases of zero dimensional ideals. In *Effective methods in algebraic geometry*. Springer, 227–234.
- [29] Yagati N Lakshman and Daniel Lazard. 1991. On the complexity of zero-dimensional algebraic systems. In *Effective methods in algebraic geometry*. Springer, 217–225.
- [30] J.B. Lasserre, M. Laurent, and P. Rostalski. 2008. Semidefinite Characterization and Computation of Zero-Dimensional Real Radical Ideals. *Foundations of Computational Mathematics* 8 (2008), 607–647. Issue 5.
- [31] J.B. Lasserre, M. Laurent, and P. Rostalski. 2009. A prolongation-projection algorithm for computing the finite real variety of an ideal. *Theoretical Computer Science* 410, 27–29 (2009), 2685–2700.
- [32] J.B. Lasserre, M. Laurent, and P. Rostalski. 2009. A unified approach to computing real and complex zeros of zero-dimensional ideals. In *Emerging applications of algebraic geometry*. IMA Vol. Math. Appl., Vol. 149. Springer, New York, 125–155.
- [33] Jean B Lasserre. 2001. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization* 11, 3 (2001), 796–817.
- [34] Monique Laurent. 2004. Revisiting Two Theorems of Curto and Fialkow on Moment Matrices. *Proc. Amer. Math. Soc.* 133 (02 2004). <https://doi.org/10.2307/4097911>
- [35] Tsung-Lin Lee, Tien-Yien Li, and Chih-Hsiung Tsai. 2008. HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing* 83, 2–3 (2008), 109–133.
- [36] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. 2023. Shuffled Linear Regression with Outliers in Both Covariates and Responses. *International Journal of Computer Vision* 131 (2023), 732–751. Issue 3. <https://doi.org/10.1007/s11263-022-01709-2>
- [37] Tien-Yien Li. 1997. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta numerica* 6 (1997), 399–436.
- [38] Hao Liang, Jingyu Lu, Manolis C. Tsakiris, and Lihong Zhi. 2023. A Field-Theoretic Approach to Unlabeled Sensing. [arXiv:2303.01175](https://arxiv.org/abs/2303.01175)
- [39] Rong Ma, T. Tony Cai, and Hongzhe Li. 2020. Optimal estimation of bacterial growth rates based on a permuted monotone matrix. *Biometrika* 108, 3 (10 2020), 693–708. <https://doi.org/10.1093/biomet/asaa082> <https://academic.oup.com/biomet/article-pdf/108/3/693/39711381/asaa082.pdf>
- [40] Yue Ma and Lihong Zhi. 2012. Computing real solutions of polynomial systems via low-rank moment matrix completion. In *Proceedings of the 37th international symposium on Symbolic and algebraic computation*. 249–256.
- [41] F.S. Macaulay. 1916. *The Algebraic Theory of Modular Systems*. Cambridge Tracts in Mathematics and Mathematical Physics, Vol. 19. Cambridge.
- [42] Manuel Marques, Marko Stojić, and João Costeira. 2009. Subspace matching: Unique solution to point matching with geometric constraints. In *2009 IEEE 12th International Conference on Computer Vision*. 1288–1294. <https://doi.org/10.1109/ICCV.2009.5459318>
- [43] H. Melánová, B. Sturmfels, and R. Winter. 2022. Recovery from Power Sums. *Experimental Mathematics* (2022), 1–10.
- [44] Alexander Morgan and Andrew Sommese. 1987. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.* 24, 2 (1987), 115–138.
- [45] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. 111–125. <https://doi.org/10.1109/SP.2008.33>
- [46] Amin Nejatbakhsh and E. Varol. 2021. Neuron matching in C. elegans with robust approximate linear regression without correspondence. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2021), 2836–2845. <https://api.semanticscholar.org/CorpusID:230102094>
- [47] Arkadi Nemirovski. 2004. Interior point polynomial time methods in convex programming. *Lecture notes* 42, 16 (2004), 3215–3224.
- [48] Luong Trung Nguyen, Junhan Kim, and Byonghyo Shim. 2019. Low-Rank Matrix Completion: A Contemporary Survey. *IEEE Access* 7 (2019), 94215–94237. <https://doi.org/10.1109/ACCESS.2019.2928130>
- [49] Jiawang Nie and Li Wang. 2014. Semidefinite relaxations for best rank-1 tensor approximations. *SIAM J. Matrix Anal. Appl.* 35, 3 (2014), 1155–1179.
- [50] Jiawang Nie, Li Wang, and Zequn Zheng. 2023. Low rank tensor decompositions and approximations. *Journal of the Operations Research Society of China* (2023), 1–27.
- [51] Pablo A Parrilo. 2000. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology.
- [52] Pablo A Parrilo. 2003. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming* 96 (2003), 293–320.
- [53] L. Peng and M. C. Tsakiris. 2020. Linear Regression Without Correspondences via Concave Minimization. *IEEE Signal Processing Letters* 27 (2020), 1580–1584.
- [54] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* 52, 3 (2010), 471–501.
- [55] Greg Reid and Lihong Zhi. 2009. Solving polynomial systems via symbolic-numeric reduction to geometric involutive form. *Journal of Symbolic Computation* 44, 3 (2009), 280–291.
- [56] Christopher Rose and I. Saira Mian. 2014. Signaling with identical tokens: Upper bounds with energy constraints. In *2014 IEEE International Symposium on Information Theory*. 1817–1821. <https://doi.org/10.1109/ISIT.2014.6875147>
- [57] M. Slawski and E. Ben-David. 2019. Linear regression with sparsely permuted data. *Electronic Journal of Statistics* 13, 1 (2019), 1–36.
- [58] Xuming Song, Hayoung Choi, and Yuanming Shi. 2018. Permuted Linear Model for Header-Free Communication via Symmetric Polynomials. In *2018 IEEE International Symposium on Information Theory (ISIT)*. 661–665. <https://doi.org/10.1109/ISIT.2018.8437552>
- [59] Jos F. Sturm. 1999. Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* 11, 1–4 (1999), 625–653. <https://doi.org/10.1080/10556789908805766>
- [60] Defeng Sun, Kim-Chuan Toh, Yancheng Yuan, and Xin-Yuan Zhao. 2019. SDPNAL+: A Matlab software for semidefinite programming with bound constraints (version 1.0). [arXiv:1710.10604](https://arxiv.org/abs/1710.10604) [math.OC]
- [61] Manolis C. Tsakiris. 2023. Matrix Recovery from Permutations: An Algebraic Geometry Approach. In *2023 IEEE International Symposium on Information Theory (ISIT)*. 2511–2516. <https://doi.org/10.1109/ISIT54713.2023.10206879>
- [62] Manolis C. Tsakiris and Liangzu Peng. 2019. Homomorphic Sensing. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6335–6344. <https://proceedings.mlr.press/v97/tsakiris19a.html>
- [63] Manolis C. Tsakiris, Liangzu Peng, Aldo Conca, Laurent Kneip, Yuanming Shi, and Hayoung Choi. 2020. An Algebraic-Geometric Approach for Linear Regression Without Correspondences. *IEEE Transactions on Information Theory* 66, 8 (2020), 5130–5144. <https://doi.org/10.1109/TIT.2020.2977166>
- [64] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. 2015. Unlabeled sensing: Solving a linear system with unordered measurements. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 786–793. <https://doi.org/10.1109/ALLERTON.2015.7447086>
- [65] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. 2018. Unlabeled Sensing With Random Linear Measurements. *IEEE Transactions on Information Theory* 64, 5 (2018), 3237–3253. <https://doi.org/10.1109/TIT.2018.2809002>
- [66] Bart Vandereycken. 2013. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization* 23, 2 (2013), 1214–1236.
- [67] Jan Verschelde. 1999. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)* 25, 2 (1999), 251–276.
- [68] Zhengfeng Yang, Hanrui Zhao, and Lihong Zhi. 2023. VerifyRealRoots: A Matlab Package for Computing Verified Real Solutions of Polynomials Systems of Equations and Inequalities. *Journal of Systems Science and Complexity* 36, 2 (2023), 866–883.
- [69] Zhengfeng Yang, Lihong Zhi, and Yijun Zhu. 2013. Verified error bounds for real solutions of positive-dimensional polynomial systems. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. 371–378.