ALGEBRAICALLY-INITIALIZED EXPECTATION MAXIMIZATION FOR HEADER-FREE COMMUNICATION

Liangzu Peng, Xuming Song, Manolis C. Tsakiris
Hayoung Choi, Laurent Kneip, and Yuamming Shi

School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
{penglz, songxm, mtsakiris, hchoi, lkneip, shiym}@shanghaitech.edu.cn

ABSTRACT

Towards low-latency communication for short-packet transmission, this paper tackles the problem of shuffled linear regression for large-scale wireless sensor networks with header-free communication by using results from algebraic geometry as well as an alternating optimization scheme. The shuffled linear regression problem is to solve a linear system with shuffled entries of the right hand side vector. However, solving the shuffled linear system requires high computational cost. The key idea of our approach is to eliminate the shuffled structure via symmetric polynomials, which leads to a system of polynomial equations. Considering one of the solutions of the resulting polynomial system as an initialization to the Expectation Maximization algorithm, we propose the Algebraically-Initialized Expectation Maximization algorithm. Computational experiments with synthetic data show that our proposed algorithm is extensively efficient, and it performs well even with noise.

Index Terms— Header-free communication, shuffled linear regression, permuted linear model, symmetric polynomials, expectation maximization, algebraic geometry.

1. INTRODUCTION

The development of the fifth-generation (5G) technologies is being accelerated by the increasing demands for new vertical services such as massive Internet of Things (IoT), ultra-reliable and low-latency communication (URLLC) services, and massive machine-to-machine communications (MM2M) [1]. As an example in URLLC services, data packets are exchanged with demanding requirements in terms of reliability (e.g., 99.999%) and latency (e.g., 4 ms). On the other hand, the 5G system will need to support novel traffic types that use short packets for emerging applications [2]. For example, in the scenario of MM2M, the data packets, exchanged among massive number of devices and containing critical measurements, are usually short, while the reliability and latency for the communication are expected to be ultrahigh and low, respectively. However, as shown in Fig. 1, the control information (metadata) is of negligible size compared to the transmitted data (payload) in the long data packet, which as an assumption of current wireless systems is not true for the short packet. In contrast, in the short packet, the metadata is even larger than the payload. As a consequence, the transmission latency may be highly suboptimal when applying traditional methods relying on this assumption to transmit short packets. To improve the latency for short-packet transmission, several attempts have been made [2, 3], among which a simple one is to exclude from the packet the header containing identity information, i.e., a bitstring that identifies the source device to the sink one [4]. This exclusion leads to the *header-free communication* [5]. Although the header-free communication reduces the packet size and hence accelerates the transmission, recovering the source signal at the sink can be challenging.

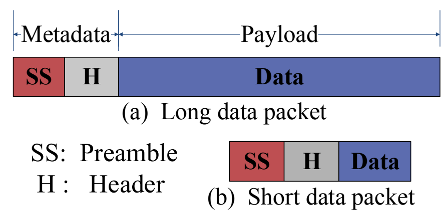

Fig. 1. Example of the long and short data packet.

1.1. System Model

Consider a massive sensor network where m sensor nodes $s_1, \dots, s_m$ sense a one-dimensional signal $x_1^*, \dots, x_n^* \in \mathbb{R}$ changed over time (indexed by $1, \dots, n$), such as temperature, pressure, or moisture, from an area of interest and send the signal to the fusion center for further processing. We may assume $m \gg n$ with n sufficiently small, e.g., $m = 50, n = 5$ [6]. Each sensor s_i obtains a corrupted measurement $x_j^* + w_{i,j}$ at time j for $j \in \{1, \dots, n\} =: [n]$, where $w_{i,j} \in \mathbb{R}$ is noise. Since IoT sensors are typically power lim-

This work was supported in part by the National Nature Science Foundation of China under Grant 61601290 and in part by the Shanghai Sailing Program under Grant 16YF1407700.

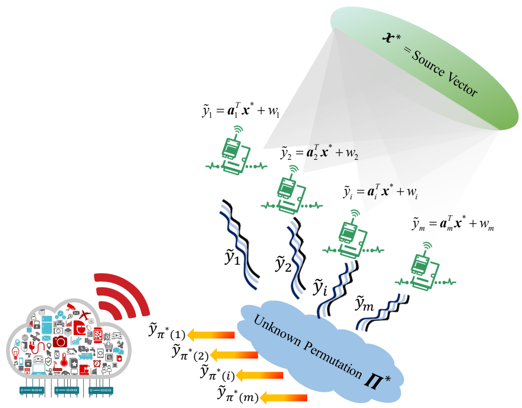

Fig. 2. System model for header-free communication.

ited and the consecutive measurements are highly correlated, it is quite reasonable for each sensor to send the weighted average $a_{i,1}(x_1+w_{i,1})+\cdots+a_{i,n}(x_n+w_{i,n})$ of its observations over time to the fusion center, weighted by $a_{i,1},\ldots,a_{i,n} \in \mathbb{R}$ with $\Sigma_{j=1}^n a_{i,j} = 1$, as a way of compressing the information and reducing the transmission overhead [7]. Finally, the center receives from each sensor s_i the averaged measurement

$$\tilde{y}_i := \boldsymbol{a}_i^\top \boldsymbol{x}^* + w_i, \tag{1}$$

where $\boldsymbol{a}_i = [a_{i,1},\ldots,a_{i,n}]^\top$, $w_i = a_{i,1}w_{i,1}+\cdots+a_{i,n}w_{i,n}$, and $\boldsymbol{x}^* = [x_1^*,\ldots,x_n^*]^\top$ is the source vector to be recovered. When the sensors send packets with the header, the fusion center understands that the observation $\tilde{y}_i$ is sent by the sensor s_i. Consequently the source vector $\boldsymbol{x}^*$ satisfies

$$\tilde{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{w}, \tag{2}$$

where $\tilde{\boldsymbol{y}} = [\tilde{y}_1,\ldots,\tilde{y}_m]^\top$, $\boldsymbol{A} = [\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m]^\top$, and $\boldsymbol{w} = [w_1,\ldots,w_m]^\top$. The linear system of equations (2) is called *fixed SNR linear observation model* in [7].

However, when the identity information is no longer available (Fig. 2), what the fusion center receives is in fact a shuffled observation vector $[\tilde{y}_{\pi^*(1)},\ldots,\tilde{y}_{\pi^*(m)}]^\top =: \boldsymbol{y}$, where π^* presents an unknown permutation, or equivalently $\boldsymbol{y} = (\boldsymbol{\Pi}^*)^\top \tilde{\boldsymbol{y}}$, where the unknown permutation matrix $\boldsymbol{\Pi}^* = [\boldsymbol{e}_{\pi^*(1)},\ldots,\boldsymbol{e}_{\pi^*(m)}]$, with $\boldsymbol{e}_i$ the i-th standard basis vector in $\mathbb{R}^m$. Consequently, the problem of recovering $\boldsymbol{x}^*$ is reduced to solving the shuffled linear system

$$\boldsymbol{\Pi}\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \tag{3}$$

for $\boldsymbol{x}$ with $\boldsymbol{\Pi}$ a unknown permutation matrix. This is essentially the problem of *shuffled linear regression* [8, 9] or *permuted linear model* [3], also known as *linear regression with sparsely permuted data* [10], *unlabeled ordered sensing* [11], *unlabeled sensing* [12], *signal processing with unlabeled data* [13], or *homomorphic sensing* [14, 15].

1.2. Prior Art

Consider the shuffled linear system (3) with no noise, i.e., $\boldsymbol{w} = \boldsymbol{0}$, where $\boldsymbol{0} \in \mathbb{R}^m$ is the zero vector. If $\boldsymbol{A}$ is *generic* [8], the authors in [12] show that the source vector $\boldsymbol{x}^*$ can be uniquely recovered with probability 1 as long as $m \geq 2n$ while $m \geq 2n$ is necessary for this guarantee. Moreover, it is proved in [8] that the unique recovery of $\boldsymbol{x}^*$ is possible even if $m < 2n$, provided that $n < m$ and $\boldsymbol{y}$ a permuted version of a generic vector in the range space of $\boldsymbol{A}$. On the other hand, some recent efforts are put on algorithmic development for the case $\boldsymbol{w} = \boldsymbol{0}$. An obvious algorithm for recovering $\boldsymbol{x}^*$ is brute force: for every possible permutation $\boldsymbol{\Pi}$ solve the linear system $\boldsymbol{\Pi}\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. This is an algorithm with exponential complexity $\mathcal{O}((m!)n^2m)$, far from being practical. Both with complexity lower than the brute force one, the geometric reconstruction algorithm proposed in [16] is of complexity $\mathcal{O}(nm^{n+1})$ and an algorithm in [17] is proposed with a complexity at least m^7 [8]. However, the geometric reconstruction algorithm can only handle noise for $n = 2$ with a carefully designed matrix $\boldsymbol{A}$ and extending it to higher dimensions is nontrivial, while the algorithm in [17] is fragile in the presence of noise. The authors in [3] propose to solve the noiseless shuffled linear system via symmetric polynomials, leading to an algorithm with complexity linear in m and exponential in n, suitable for the case $m \gg n$ with n small. However, no algorithm in [3] is proposed for the noisy case.

Another common approach towards solving (3) is by computing the *Maximal Likelihood Estimator* (MLE)

$$(\hat{\boldsymbol{x}}_{\text{ML}}, \widehat{\boldsymbol{\Pi}}_{\text{ML}}) = \underset{\boldsymbol{\Pi}\in\mathcal{S}_m, \boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \left\|\boldsymbol{\Pi}\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\right\|_2, \tag{4}$$

where $\mathcal{S}_m$ is the set of all $m \times m$ permutation matrices. Assuming that $\boldsymbol{A}$ is drawn from the standard normal distribution, the behavior of the MLE $(\hat{\boldsymbol{x}}_{\text{ML}}, \widehat{\boldsymbol{\Pi}}_{\text{ML}})$ in terms of the SNR is studied in [5, 17, 18]. The authors in [5] show that with high probability $\widehat{\boldsymbol{\Pi}}_{\text{ML}}$ coincides with $\boldsymbol{\Pi}^*$ when SNR $\geq m^{c_1}$ for c_1 some absolute constant. If however SNR $< c_2 \min\{1, n\log(\log(m))\}$ for some $c_2 > 0$, then no estimator, including $\hat{\boldsymbol{x}}_{\text{ML}}$, can approximate $\boldsymbol{x}^*$ provided $m \geq 3$ and $n \geq 22$ [17]. Finally, the authors in [18] show that when the SNR is fixed the ML estimator $\hat{\boldsymbol{x}}_{\text{ML}}$ is inconsistent. On the other hand, although the MLE is NP-hard to compute when $n > 1$ [5], the authors in [17] propose a $(1+\epsilon)$ approximation algorithm of complexity $\mathcal{O}((m/\epsilon)^n)$. Some works [9, 19, 20] compute (4) via alternating minimization. This approach alternatively sorts a vector (in $\mathcal{O}(m\log(m))$ time) and solves a linear system (in $\mathcal{O}(n^2m)$ time) in each iteration and thus enjoys an acceptable complexity, it however tends to fail unless a good initialization is provided.

1.3. Contribution

In this paper we make progress towards efficiently computing the MLE (4) for the shuffled linear system with noise (3) via

Algebraically-Initialized Expectation Maximization, abbreviated as the AI-EM algorithm (§2) [8]. Although the complexity of the proposed algorithm is exponential in n, it is linear in m (the number of sensors in our discussion). In addition, the algorithm is able to deal with noise. Hence, it is suitable for the shuffled linear regression problem in the scenario of IoT sensor networks with header-free communication, where typically m is large (e.g., $m = 500$) and n is small (e.g., $n = 4$).

2. ALGEBRAICALLY-INITIALIZED EXPECTATION MAXIMIZATION

In this section we present the AI-EM algorithm [8]. First, we introduce a polynomial system derived from shuffled linear system with noise (3) via symmetric polynomials. With the theoretical guarantee that the derived polynomial system admits finitely many solutions [8], we solve it and obtain finitely many roots. Finally, we extract from the roots the most suitable one and use it as an initialization to the Expectation Maximization algorithm.

2.1. Symmetric Polynomials and the Polynomial System

To begin with, let $\mathbb{R}[\boldsymbol{z}] := \mathbb{R}[z_1, \dots, z_m]$ be the set of polynomials with real coefficients over variables $\boldsymbol{z} := [z_1, \dots, z_m]$. A polynomial $p(\boldsymbol{z}) := p(z_1, \dots, z_m) \in \mathbb{R}[\boldsymbol{z}]$ is *symmetric* if it is invariant to any permutation of the variables $\boldsymbol{z}$, i.e.,

$$p(\boldsymbol{z}) = p(z_{\pi(1)}, \dots, z_{\pi(m)}) =: p(\boldsymbol{\Pi z}), \tag{5}$$

where π is a permutation on $[m]$ and $\boldsymbol{\Pi} \in \mathbb{R}^{m \times m}$ is the corresponding permutation matrix. The power-sum polynomial

$$p_k(\boldsymbol{z}) = \Sigma_{i=1}^m z_i^k, \tag{6}$$

for example, satisfies (5) for k some non-negative integer, and thus is symmetric.

As shown in [3], symmetric polynomials are used to eliminate the permutation matrix $\boldsymbol{\Pi}$ in (3). To see this, let $\boldsymbol{A}, \boldsymbol{y}, \boldsymbol{w}$ be as in (3) and $p_{\text{sym}} \in \mathbb{R}[\boldsymbol{z}]$ a symmetric polynomial in $\mathbb{R}[\boldsymbol{z}]$. Then given any solution $(\boldsymbol{\Pi}^*, \boldsymbol{x}^*)$ for (3), we have

$$\begin{aligned} &p_{\text{sym}}(\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{w}) = p_{\text{sym}}(\boldsymbol{\Pi}^*\boldsymbol{y}) = p_{\text{sym}}(\boldsymbol{y}) \\ \iff &p_{\text{sym}}(\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{w}) - p_{\text{sym}}(\boldsymbol{y}) = 0. \end{aligned} \tag{7}$$

Specifically, every solution for (3) needs to satisfy n polynomial equations in n variables $x_1, \dots, x_n$

$$\tilde{p}_k(\boldsymbol{x}) := p_k(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}) - p_k(\boldsymbol{y}) = 0, k \in [n], \tag{8}$$

where p_k is the power-sum polynomial as defined in (6). Since the vector $\boldsymbol{w}$ in (8) is unknown, the solutions of the polynomial system $\tilde{\mathcal{P}}$ given by $\tilde{p}_1(\boldsymbol{x}) = \cdots = \tilde{p}_n(\boldsymbol{x}) = 0$ are not available. Instead we resort to an alternative polynomial system $\hat{\mathcal{P}}$,

$$\hat{p}_k(\boldsymbol{x}) := p_k(\boldsymbol{A}\boldsymbol{x}) - p_k(\boldsymbol{y}) = 0, k \in [n]. \tag{9}$$

Then since $\boldsymbol{A}, \boldsymbol{y}$ are both known, the polynomial system $\hat{\mathcal{P}}$ contains n equations in n variables $x_1, \dots, x_n$. However, does the system $\hat{\mathcal{P}}$ admit any solutions? If yes then are there finitely many? Quite importantly, [8] has established that as long as A is generic, the solution set $\mathcal{V} := \{\hat{\boldsymbol{x}} \in \mathbb{C}^n : \hat{p}_k(\hat{\boldsymbol{x}}) = 0, \forall k \in [n]\}$ of the polynomial system $\hat{\mathcal{P}}$ is almost always non-empty and finite, containing at most $n!$ number of points. Consequently, the system $\hat{\mathcal{P}}$ is readily solvable by standard algorithms in numerical algebraic geometry. One may expect at least one real root of $\hat{\mathcal{P}}$ to be a reasonably good approximation to the solution for our original problem (3).

2.2. Algebraically-Initialized Expectation Maximization

Algorithm 1 Algebraically-Initialized Expectation Maximization

procedure AI-EM($\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $T \in \mathbb{N}$)
  $\{\hat{\boldsymbol{x}}_i\}_{i=1}^L \leftarrow$ solutions of $\hat{\mathcal{P}}$ (9);
  $\{(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}\}_{i=1}^L \leftarrow \{a : a + ib \in \{\hat{\boldsymbol{x}}_i\}_{i=1}^L\}$;
  $\hat{\boldsymbol{x}}_0 \leftarrow \text{argmin}_{i \in [L]} \{\min_{\boldsymbol{\Pi}} \|\boldsymbol{\Pi y} - \boldsymbol{A}(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}\|_2\}$;
  for $t = 1 : T$ **do**
    $\boldsymbol{\Pi}_t \leftarrow \text{argmin}_{\boldsymbol{\Pi}} \|\boldsymbol{\Pi y} - \boldsymbol{A}\boldsymbol{x}_{t-1}\|_2$;
    $\boldsymbol{x}_t \leftarrow \text{argmin}_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{\Pi}_t \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$;
  end for
  Return $\boldsymbol{x}_t$.
end procedure

As discussed in §2.1, we can solve the polynomial system $\hat{\mathcal{P}}$ and obtain roots $\{\hat{\boldsymbol{x}}_i\}_{i=1}^L$ with $0 < L \leq n!$. Since the polynomial system $\hat{\mathcal{P}}$ is zero-dimensional [8], solving it requires a double-exponential or exponential complexity in n, depending on the algorithms used [21]. On the other hand, it demands a complexity linear in m because of the matrix-vector product $\boldsymbol{A}\boldsymbol{x}$.

Because the roots $\{\hat{\boldsymbol{x}}_i\}_{i=1}^L$ obtained from solving the system $\hat{\mathcal{P}}$ are potentially complex and the source vector $\boldsymbol{x}^*$ is in $\mathbb{R}^n$, we only work with the real parts $\{(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}\}_{i=1}^L$ of the roots $\{\hat{\boldsymbol{x}}_i\}_{i=1}^L$. From these roots we select the estimate $\hat{\boldsymbol{x}}_{\text{AI}}$ that best approximates the ML estimator $\hat{\boldsymbol{x}}_{\text{ML}}$, i.e.,

$$\hat{\boldsymbol{x}}_{\text{AI}} := \underset{i \in [L]}{\text{argmin}} \left\{ \min_{\boldsymbol{\Pi}} \|\boldsymbol{\Pi y} - \boldsymbol{A}(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}\|_2 \right\}. \tag{10}$$

We note that by the rearrangement inequality of [22], the inner minimization $\min_{\boldsymbol{\Pi}} \|\boldsymbol{\Pi y} - \boldsymbol{A}(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}\|_2$ in (10) is achieved when $\boldsymbol{\Pi}$ is the permutation matrix that permutes the entries of $\boldsymbol{y}$ according to the order of the elements of $\boldsymbol{A}(\hat{\boldsymbol{x}}_i)_{\mathbb{R}}$, and hence the minimization problem can be solved via sorting [9], a procedure of complexity $\mathcal{O}(m \log(m))$. Also noting that the number L of estimates is at most $n!$ [8], the computation of $\hat{\boldsymbol{x}}_{\text{AI}}$ is of complexity $\mathcal{O}((n!) m \log(m))$.

Going further, we use the *algebraic initialization* $\hat{\boldsymbol{x}}_{\text{AI}}$ as an initialization to the Expectation Maximization (EM) algorithm in [9], which alternatively updates $\boldsymbol{\Pi}$ and $\boldsymbol{x}$. The update of $\mathbf{\Pi}$ is done by sorting in $\mathcal{O}(m\log(m))$ time, while updating $\boldsymbol{x}$ involves solving a linear system, entailing a complexity of $\mathcal{O}(n^2 m)$. Hence this EM algorithm is of complexity $\mathcal{O}(T(m\log(m)+n^2 m))$ with T the number of iterations. The complete procedure mentioned above, referred as to *Algebraically-Initialized Expectation Maximization (AI-EM)*, is presented in Algorithm 1.

3. SIMULATION RESULTS

Methods. We evaluate the performance of the AI-EM algorithm, and we also provide a comparison with another EM-like algorithm proposed in [9] and the robust regression algorithm in [10]. The algorithm in [9] computes the MLE via alternating minimization exactly as in Algorithm 1, except that it initializes the computation by the least-squares solution

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2. \tag{11}$$

For both AI-EM and the algorithm in [9] we set the number T of iterations to be 100. On the other hand, the algorithm in [10] treats permuted data as outliers and uses a robust regression formulation to estimate the source vector $\boldsymbol{x}^*$.

Data. The experiments for the shuffled linear system with noise (3) rely on synthetic data generated randomly in the following way. The noise vector $\boldsymbol{w}\in\mathbb{R}^n$ is sampled from the zero-mean distribution with covariance matrix $\sigma^2\boldsymbol{I}_m$, while the first $n-1$ columns of $\boldsymbol{A}$ and the source vector $\boldsymbol{x}^*\in\mathbb{R}^n$ are drawn from corresponding standard normal distributions. The last column of $\boldsymbol{A}$ is set to a vector so that the sum of each row of $\boldsymbol{A}$ is 1. With $\boldsymbol{w}$, $\boldsymbol{A}$ and $\boldsymbol{x}^*$ comes the corrupted observation $\tilde{\boldsymbol{y}}:=\boldsymbol{A}\boldsymbol{x}^*+\boldsymbol{w}$. We further permute $\tilde{\boldsymbol{y}}$ by left-multiplying a matrix $(\mathbf{\Pi}^*)^\top$ uniformly drawn from $\mathcal{P}_m$. In this way we have obtained as inputs the data $\boldsymbol{A}$ and $(\mathbf{\Pi}^*)^\top\tilde{\boldsymbol{y}}=:\boldsymbol{y}$ for the algorithms to be evaluated.

Metrics. We report the relative estimation error of $\boldsymbol{x}^*$ for the estimate $\hat{\boldsymbol{x}}$, computed by $100\times\frac{\|\boldsymbol{x}^*-\hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}^*\|_2}\%$.

Results. Fig. 3 shows the relative estimation error of two algorithms for SNR $=0:10:60$ with $n=4$ and $m=500$. In agreement with the results in [17], the estimation error of the AI-EM algorithm (red) is large when the SNR is small. As the SNR increases, the error decreases and tends to converge, finally achieving an estimation error of 0.4% for SNR $=60$. On the other hand, the algorithm (blue) in [9] fails since the resulting estimation errors are consistently larger than 100% and the same happens to the algorithm (black) in [10]. This suggests that the least-squares initialization for alternating minimization may not be a good choice when data are fully shuffled [8], and that the assumption of sparsely permuted data in [10] is essential for their algorithm to work [15]. However, the scenario where data is fully shuffled is exactly the case in header-free communication since all sensors tacitly send packets without identity information which then arrive at the fusion center in an arbitrary order.

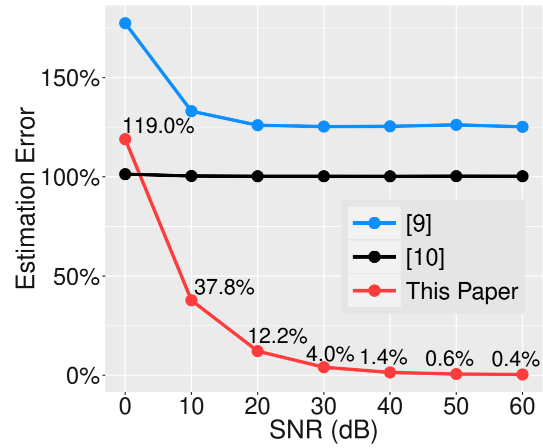

Fig. 3. Estimation error for varying SNR with $m=500, n=4$ fixed, averaged over 10000 independent trials.

We also report the running times of the AI-EM algorithm with $m=500$, SNR $=30$ fixed and n varying from 3 to 6. To solve the polynomial system $\hat{\mathcal{P}}$, we use our custom Gröbner basis solvers automatically generated based on elimination template techniques [23] for $n=3,4$, and a standard homotopy-based solver, Bertini [24], for $n=5,6$. On a PC (Intel i7-8650, 1.9GHz, 16GB), the AI-EM algorithm runs in 5 milliseconds for $n=3$, 11 milliseconds for $n=4$, 45 seconds for $n=5$, and 47 minutes for $n=6$. The efficiency for $n=3,4$ is on account of the solver generator utilizing the special structure of the polynomial system $\hat{\mathcal{P}}$, while the exponentially increasing running time for $n=5,6$ is partly because the general-purpose solver Bertini is slower than the custom one, and partly because the polynomial system is becoming exponentially complicated.

4. CONCLUSION

We presented a working solution to shuffled linear regression for massive IoT sensor networks with header-free communication. The key algorithmic idea was to solve a polynomial system derived from the original problem to obtain some solutions and then extract from the solutions the most suitable one as initialization to the alternating minimization algorithm. The proposed algorithm was shown to be suitable for this problem since its complexity is linear in the number of sensors. Moreover, not limited to sensor networks with header-free communication, the proposed algorithm can be widely applied to other applications such as simultaneous pose and correspondence estimation [25], multi-target tracking [26] in computer vision and so on.

5. REFERENCES

[1] K. S. Kim, D. K. Kim, C. Chae, S. Choi, Y. Ko, J. Kim, Y. Lim, M. Yang, S. Kim, B. Lim, K. Lee, and K. L. Ryu, "Ultrareliable and low-latency communication techniques for tactile internet services," *Proc. IEEE*, pp. 1–18, 2018.

[2] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sept., 2016.

[3] X. Song, H. Choi, and Y. Shi, "Permuted linear model for header-free communication via symmetric polynomials," in *2018 IEEE Int. Symp. Inf. Theory (ISIT)*, June, 2018, pp. 661–665.

[4] L. Keller, M. J. Siavoshani, C. Fragouli, K. Argyraki, and S. Diggavi, "Identity aware sensor networks," in *IEEE INFOCOM 2009*, Apr., 2009, pp. 2177–2185.

[5] A. Pananjady, M. J. Wainwright, and T. A. Courtade, "Linear regression with shuffled data: Statistical and computational limits of permutation recovery," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3286–3300, May, 2018.

[6] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. 4th Int. Symp. Inf. Process. Sensor Netw., 2005.*, April 2005, pp. 63–70.

[7] S. Aeron, M. Zhao, and V. Saligrama, "On sensing capacity of sensor networks for a class of linear observation models," in *2007 IEEE/SP 14th Workshop Stat. Signal Process.*, Aug 2007, pp. 388–392.

[8] M. C. Tsakiris, L. Peng, A. Conca, L. Kneip, Y. Shi, and H. Choi, "An Algebraic-Geometric Approach to Shuffled Linear Regression," *ArXiv e-prints*, Oct., 2018.

[9] A. Abid and J. Zou, "Stochastic EM for Shuffled Linear Regression," *ArXiv e-prints*, Apr., 2018.

[10] M. Slawski and E. Ben-David, "Linear regression with sparsely permuted data," *Electron. J. Stat.*, vol. 13, no. 1, pp. 1–36, 2019.

[11] S. Haghighatshoar and G. Caire, "Signal recovery from unlabeled samples," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1242–1257, 2018.

[12] J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli, "Unlabeled sensing with random linear measurements," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3237–3253, May, 2018.

[13] S. Marano and P. Willett, "Algorithms and Fundamental Limits for Unlabeled Detection using Types," *ArXiv e-prints*, Oct., 2018.

[14] M. C. Tsakiris, "Eigenspace conditions for homomorphic sensing," *ArXiv e-prints*, Dec., 2018.

[15] M. C. Tsakiris and L. Peng, "Homomorphic sensing," *ArXiv e-prints*, Jan., 2019.

[16] G. Elhami, A. Scholefield, B. Béjar Haro, and M. Vetterli, "Unlabeled sensing: Reconstruction algorithm and theoretical guarantees," in *IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, Mar., 2017, pp. 4566–4570.

[17] D. J. Hsu, K. S. Shi, and X. Sun, "Linear regression without correspondence," in *Advances Neural Inf. Proc. Syst. 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 1531–1540. Curran Associates, Inc., 2017.

[18] A. Abid, A. Poon, and J. Zou, "Linear Regression with Shuffled Labels," *ArXiv e-prints*, May, 2017.

[19] J. Zhu, H. Cao, C. Song, and Z. Xu, "Parameter estimation via unlabeled sensing using distributed sensors," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2130–2133, Oct., 2017.

[20] G. Wang, J. Zhu, R. S. Blum, P. Willett, S. Marano, V. Matta, and P. Braca, "Signal amplitude estimation and detection from unlabeled binary quantized samples," *IEEE Trans. Signal Process.*, vol. 66, no. 16, pp. 4291–4303, Aug., 2018.

[21] A. Ayad, "A survey on the complexity of solving algebraic systems," in *Int. Math. Forum*, 2010, vol. 5, pp. 333–353.

[22] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities (Cambridge Mathematical Library)*, Cambridge University Press, 1988.

[23] L. Kneip, "Polyjam," `https://github.com/laurentkneip/polyjam`, 2015.

[24] D. J. Bates, J. D. Hauenstein, S. J. Andrew, and C.W. Wampler., "Bertini: Software for numerical algebraic geometry," Available at bertini.nd.edu.

[25] P. David, D. Dementhon, R. Duraiswami, and Hanan H. Samet, "Softposit: Simultaneous pose and correspondence determination," *Int. J. Comput. Vis.*, vol. 59, no. 3, pp. 259–284, 2004.

[26] A. B. Poore and S. Gadaleta, "Some assignment problems arising from multiple target tracking," *Math. Comput. Model.*, vol. 43, no. 9-10, pp. 1074–1091, May, 2006.