**Fifty-Fifth Annual Allerton Conference**
**Allerton House, UIUC, Illinois, USA**
**October 3-6, 2017**

# Learning Mixtures of Sparse Linear Regressions Using Sparse Graph Codes

Dong Yin*, Ramtin Pedarsani†, Yudong Chen‡, Kannan Ramchandran*
*Department of Electrical Engineering and Computer Sciences, UC Berkeley
{dongyin, kannanr}@eecs.berkeley.edu
†Department of Electrical and Computer Engineering, UC Santa Barbara
ramtin@ece.ucsb.edu
‡School of Operations Research and Information Engineering, Cornell University
yudong.chen@cornell.edu

***Abstract*—In this paper, we consider the *mixture of sparse linear regressions* model. Let $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(L)} \in \mathbb{C}^n$ be $L$ unknown sparse parameter vectors with a total of $K$ non-zero coefficients. Noisy linear measurements are obtained in the form $y_i = \boldsymbol{x}_i^{\mathrm{H}} \boldsymbol{\beta}^{(\ell_i)} + w_i$, each of which is generated randomly from one of the sparse vectors with the label $\ell_i$ unknown. The goal is to estimate the parameter vectors efficiently with low sample and computational costs. This problem presents significant challenges as one needs to simultaneously solve the *demixing* problem of recovering the labels $\ell_i$ as well as the *estimation* problem of recovering the sparse vectors $\boldsymbol{\beta}^{(\ell)}$.**

**Our solution to the problem leverages the connection between modern coding theory and statistical inference. We introduce a new algorithm, *Mixed-Coloring*, which samples the mixture strategically using query vectors $\boldsymbol{x}_i$ constructed based on ideas from sparse graph codes. Our novel code design allows for both efficient demixing and parameter estimation. The algorithm achieves the order-optimal sample and time complexities of $\Theta(K)$ in the noiseless setting, and near-optimal $\Theta(K \operatorname{polylog}(n))$ complexities in the noisy setting. In one of our experiments, to recover a mixture of two regressions with dimension $n = 500$ and sparsity $K = 50$, our algorithm is more than $300$ times faster than EM algorithm, with about $1/3$ of its sample cost.**

## I. Introduction

Mixture and latent variable models, such as Gaussian mixtures and subspace clustering, are expressive, flexible, and widely used in a broad range of problems including background modeling [1], speaker identification [2] and recommender systems [3]. However, parameter estimation in mixture models is notoriously difficult due to the non-convexity of the likelihood functions and the existence of local optima. In particular, it often requires a large sample size and many re-initializations of the algorithms to achieve an acceptable accuracy.

Our goal is to develop provably fast and efficient algorithms for mixture models — with sample and time complexities *sublinear* in the problem's ambient dimension when the parameter vectors of interest is sparse — by leveraging the underlying low-dimensional structures.

In this paper we focus on a powerful class of models called *mixtures of linear regressions* [4]. We consider the *sparse* setting with a *query-based* algorithmic framework. In particular, we assume that each query-measurement pair $(\boldsymbol{x}_i, y_i)$ is generated from a sparse linear model chosen randomly from $L$ possible models:[1]

$$y_i = \boldsymbol{x}_i^{\mathrm{H}} \boldsymbol{\beta}^{(\ell)} + w_i \quad \text{with probability } q_\ell, \quad \text{for } \ell \in [L], \quad (1)$$

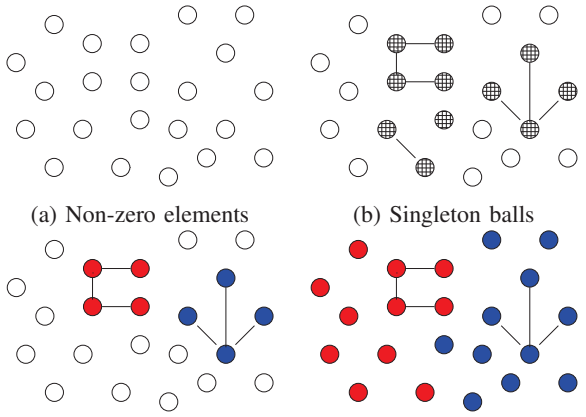
where $w_i$ is noise. The total number of nonzero elements in the parameter vectors $\{\boldsymbol{\beta}^{(\ell)} \in \mathbb{C}^n, \ell \in [L]\}$ is assumed to be $K$. The goal is to estimate the $\boldsymbol{\beta}^{(\ell)}$'s, without knowing which $\boldsymbol{\beta}^{(\ell)}$ generates each query-measurement pair.

A mixture of regressions provides a flexible model for various heterogeneous settings where the regression coefficients differ for different subsets of observations. This model has been applied to a broad range of tasks including medicine measurement design [5], behavioral health care [6] and music perception modeling [7]. Here, we study the problem when the query vectors $\boldsymbol{x}_i$ can be *designed* by the user; in Section I-B we discuss several practical applications that motivate the study of this query-based setting. Our results show that by appropriately exploiting this design freedom, one can achieve significant reduction the sample and computational costs.

To recover $K$ unknown non-zero elements, it is clear that the amount of measurements and time required scale at least as $\Theta(K)$. We introduce a new algorithm, called the *Mixed-Coloring* algorithm, that *matches these sublinear sample and time complexity lower bounds*. The design of query vectors and decoding algorithm leverages ideas from sparse graph codes such as low-density parity-check (LDPC) codes [8]. Our algorithm recovers the parameter vectors with optimal $\Theta(K)$ sample and time complexities in the noiseless setting, both in theory and empirically, and is stable under noise with near-optimal $\Theta(K \operatorname{polylog}(n))$ sample and time complexities. Prior literature on this problem that does not utilize the design freedom typically have sample/time complexities that are at least polynomial in $n$; we provide a survey of prior work and a more detailed comparison in Section VI. Empirically, we find that our algorithm is orders

[1] We use $\boldsymbol{x}_i^{\mathrm{H}}$ to denote the conjugate transpose of $\boldsymbol{x}_i$, and $[L]$ the set of integers $\{1, 2, \ldots, L\}$.

**978-1-5386-3266-6/17/$31.00 ©2017 IEEE**



(a) Non-zero elements (b) Singleton balls

(c) Giant components (d) Results

Fig. 1: Mixed-Coloring algorithm with $L = 2$.

of magnitude faster than standard Expectation-Maximization (EM) algorithms for mixture of regressions. For example, in one of our experiments, detailed in Section V, we consider recovering a mixture of two regressions with dimension $n = 500$ and sparsity $K = 50$; our algorithm is more than 300 times faster than EM algorithm, with about $1/3$ of its sample cost.

### A. Algorithm Overview

Our Mixed-Coloring algorithm solves two problems simultaneously: (i) rapiddemixing, namely identifying the *label* $\ell_i$ of the vector $\boldsymbol{\beta}^{(\ell_i)}$ that generates each measurement $y_i$; (ii) efficient identification of the *location* and *value* of the non-zero elements of the $\boldsymbol{\beta}^{(\ell)}$'s. The main idea is to use a divide-and-conquer approach that iteratively reduce the original problem into simpler ones with much sparser parameter vectors. More specifically, we design $\Theta(K)$ sets of sparse query vectors, with each set only associated with a subset of all the non-zero elements. The design of the query vectors ensures that we can first identify the sets which are associated with a single non-zero element (called singletons), and recover the location and value of that element (we call them singleton balls, shown as shaded balls in Figure 1b). We further identify the pairs of singleton balls which have the same (but unknown) label, indicated by the edges in Figure 1b. Results from random graph theory guarantees that, with high probability, the $L$ largest connected components (giant components) of the singleton graph have the different labels, and thus we recover a fraction of the non-zero elements in each $\boldsymbol{\beta}^{(\ell)}$, as shown in Figure 1c. We can then iteratively enlarge the recovered fraction with a guess-and-check method until finding all the non-zero elements. We revisit Figure 1 when describing the details of our algorithm in Section III.

### B. Motivation

Our problem is a natural extension of the setting of compressive sensing,[2] in which one often has full freedom of designing query vectors in order to estimate a sparse parameter vector. In many applications, the unknown sparse parameter vector can be affected by latent variables, leading to a mixture of sparse linear regressions, and these scenarios have been observed in neuroscience [9], genetics [10], psychology [5], etc. Here, we provide a concrete example motivated by neuroscience applications [9]. In neural signal processing, sensors are used to measure the brain activities, represented by an unknown sparse vector $\boldsymbol{\beta}$. The sensors can be modeled as digital filters, and one can *design* the linear filter weights ($\boldsymbol{x}_i$'s) when measuring the neural signal. Multiple sensors are usually placed in a particular area of the brain in order to acquire enough compressed measurements. However, there may be more than one neuron affecting a particular area of the brain, as shown in Figure 2, and each neuron may have different activities, corresponding to a different $\boldsymbol{\beta}^{(\ell)}$. Consequently, each sensor may be measuring one of several different sparse signals, which can be formulated as a mixture-of-sparse-linear-regressions problem. Variants of this problem, such as neural spike sorting [9], has been studied in neuroscience. While the common solution is to use clustering algorithms on the spike signals, we believe that our algorithm provides the potential of improving sensor design and reducing sample and time complexities.

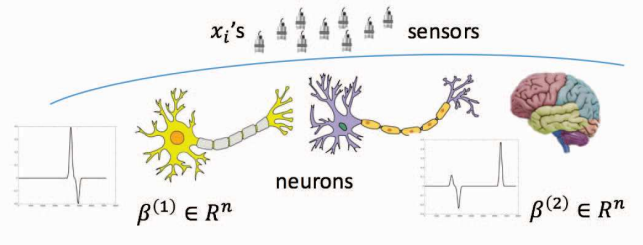



Fig. 2: Mixture of neural signals.

In addition, our work adds the intellectual value of the power of design freedom in tackling sparse mixture problems by highlighting the huge performance gap between algorithms that can exploit the design freedom and those that cannot. We also believe that our ideas are applicable more broadly for other latent-variable problems that require experimental designs, such as survey designs in psychology with mixed type of respondents and biology experiments with mixed cell interior environments.

## II. Main Results

In this section, we present the recovery guarantees for the Mixed-Coloring algorithm, and provide bounds on its sample and time complexities. We assume there are $L$ unknown $n$-dimensional parameter vectors $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(L)}$. Each $\boldsymbol{\beta}^{(\ell)}$ has $K_\ell$ non-zero elements, i.e., $|\text{supp}(\boldsymbol{\beta}^{(\ell)})| = |\{j : \beta_j^{(\ell)} \neq 0\}| = K_\ell$. Let $K = \sum_{\ell=1}^{L} K_\ell$ be the total number of non-zero elements. Using the query vectors $\{\boldsymbol{x}_i\} \in \mathbb{C}^n$, the Mixed-Coloring algorithms obtains $m$ measurements $y_i$, $i \in [m]$ generated independently according to the model (1), and outputs an estimate $\{\hat{\boldsymbol{\beta}}^{(\ell)}, \ \ell \in [L]\}$ of the unknown

[2]Compressive sensing is a special case of our problem with $L = 1$.

parameter vectors. We defer more details to Sections III and IV.

Our results are stated in the asymptotic regime where $n$ and $K$ approach infinity. A constant is a quantity that does not depend on $n$ and $K$, with the associated Big-O notations $\mathcal{O}(\cdot)$ and $\Theta(\cdot)$. We assume that $L$ is a known and fixed constant, and the mixture weights satisfy $q_\ell = \Theta(1)$ for each $\ell \in [L]$ and thus are of the same order. Similarly, the sparsity levels of the parameter vectors are also of the same order with $K_\ell = \Theta(K)$.

### A. Guarantees for the Noiseless Setting

In the noiseless case, i.e., $w_i \equiv 0$, we consider for generality the complex-valued setting with $\boldsymbol{\beta}^{(\ell)} \in \mathbb{C}^n$ (our results can be easily applied to real case).

We make a mild technical assumption, which stipulates that if any pair of parameter vectors have overlapping support, then the elements in the overlap are different.

**Assumption 1.** *For each pair $\ell_1, \ell_2 \in [L]$, $\ell_1 \neq \ell_2$ and each index $j \in \mathrm{supp}(\boldsymbol{\beta}^{(\ell_1)}) \cap \mathrm{supp}(\boldsymbol{\beta}^{(\ell_2)})$, we have $\beta_j^{(\ell_1)} \neq \beta_j^{(\ell_2)}$.*

Under the above setting, we have the following recovery guarantees for the Mixed-Coloring algorithm.

**Theorem 1.** *Consider the asymptotic regime where $n$ and $K$ approach infinity. Under Assumption 1, for any fixed constant $p^* \in (0,1)$, there exists a constant $C > 0$ such that if the number of measurements is $m = CK$, then the Mixed-Coloring algorithm satisfies the following three properties for each $\ell \in [L]$ (up to a label permutation):*

1) *(No False Discovery) $\forall\ j \in \mathrm{supp}(\boldsymbol{\beta}^{(\ell)})$, $\hat{\beta}_j^{(\ell)}$ equals either $\beta_j^{(\ell)}$ or $0$; $\forall\ j \notin \mathrm{supp}(\boldsymbol{\beta}^{(\ell)})$, $\hat{\beta}_j^{(\ell)} = 0$.*
2) *(Element-wise Recovery) There exists a constant $\tilde{p}_\ell \in (0, p^*)$ such that $\mathbb{P}\{\hat{\beta}_j^{(\ell)} = \beta_j^{(\ell)}\} = 1 - \tilde{p}_\ell - \mathcal{O}(1/K)$, $\forall\ j \in \mathrm{supp}(\boldsymbol{\beta}^{(\ell)})$.*
3) *(Support Recovery)*

$$\mathbb{P}\big\{|\mathrm{supp}(\hat{\boldsymbol{\beta}}^{(\ell)})| \geq (1-p^*)|\mathrm{supp}(\boldsymbol{\beta}^{(\ell)})|\big\} = 1 - \mathcal{O}(1/K).$$

*Moreover, the computational time of the Mixed-Coloring algorithm is $\Theta(K)$.*

The theorem ensures that the Mixed-Coloring algorithm has no false discovery, and recovers $(1-p^*)$ fraction of the non-zero elements with high probability. The error fraction $p^*$ is an input parameter to algorithm, and can be made arbitrarily close to zero by adjusting the oversampling ratio $C \equiv C(p^*, L, \{q_\ell\})$. (By more careful analysis, one can show that the dependence of $C$ on $p^*$ is $C = \mathcal{O}(\log(1/p^*))$. Here, since we set $p^*$ as a constant, $C$ is a constant.) Given the number of components $L$, mixture weights $\{q_\ell\}$ and the target $p^*$, the value of the constant $C$ can be computed numerically. The table below gives some of the $C$ values for several $p^*$ and $L$, under the setting $q_\ell = 1/L, \forall \ell \in [L]$. We see that the value of $C$ is quite modest.

We can in fact boost the above guarantee to recover all the non-zero elements, by running the Mixed-Coloring algorithm $\Theta(\log K)$ times independently and aggregating the results by majority voting. By property 2 in Theorem 1 and a union bound argument, this procedure *exactly* recovers all the parameter vectors with probability $1 - \mathcal{O}(1/\mathrm{poly}(K))$ with $\Theta(K \log K)$ sample and time complexities.

TABLE I: Sample complexity of the Mixed-Coloring algorithm

| $L$ | 2 | 3 | 4 |
|---|---|---|---|
| $p^*$ | $5.1 \times 10^{-6}$ | $8.8 \times 10^{-6}$ | $8.1 \times 10^{-6}$ |
| $m = CK$ | $33.39K$ | $37.80K$ | $40.32K$ |

### B. Guarantees for the Noisy Setting

An extension of the previous algorithm, *Robust Mixed-Coloring*, handles noise in the measurement model (1). Here we focus on the case with two parameter vectors which appear equally likely, i.e., $L = 2$ and $q_\ell = 1/2$, $\ell = 1, 2$. Many interesting applications have binary latent factors: gene mutation present/not, gender, healthy/sick individual, children/adult, etc. The noise $w_i$ is assumed to be i.i.d. Gaussian with mean zero and constant variance $\sigma^2$. For the purpose of theoretical analysis, we assume that the non-zero elements in the parameter vectors take value in a finite quantized set.

**Assumption 2.** *The non-zero elements of the parameter vectors satisfy $\beta_j^{(\ell)} \in \mathbb{D}, \forall \beta_j^{(\ell)} \neq 0, \ell \in [L]$, where*

$$\mathbb{D} \triangleq \{\pm\Delta, \pm 2\Delta, \ldots, \pm b\Delta\} \subset \mathbb{R},$$

*The positive constants $\Delta$ and $b$ are known to the algorithms.*

As shown in our empirical results in Section V, the Robust Mixed-Coloring algorithm works even when the assumption is violated. In this case, the algorithm produces the best quantized approximation to the unknown parameter vectors, provided that they are not too far off the quantized set. The theoretical results for the continuous alphabet setting is still an open problem, and the tools in recent work such as [11] may be applied to our problem.

When the quantization assumption holds, exact recovery is possible, as guaranteed in the theorem below. The Robust Mixed Coloring algorithm maintains sublinear sample and time complexities, and recovers the parameter vectors in the presence of noise with bounded variance.

**Theorem 2.** *Consider the asymptotic regime where $K$ and $n$ approach infinity with $K = \Theta(n^\alpha)$ for some constant $\alpha \in (0, 1]$. When $L = 2$ and Assumptions 1 and 2 hold, there exists a constant $\eta > 0$, such that if $\Delta/\sigma > \eta$ and the number of measurements is $m = \Theta(K \,\mathrm{polylog}(n))$, then the Robust Mixed-Coloring algorithm satisfies the three properties in Theorem 1. Moreover, the computational time of the Robust Mixed-Coloring algorithm is $\Theta(K \,\mathrm{polylog}(n))$.*

Similar to the noiseless case, by running the Robust Mixed-Coloring algorithm $\Theta(\log K)$ times, one can exactly recover the two parameter vectors with probability $1 - \mathcal{O}(1/\mathrm{poly}(K))$. In this case, the sample and computational complexities are $\Theta(K \log(K) \,\mathrm{polylog}(n))$, and further, since

we assume that $K = \Theta(n^\alpha)$ for some constant $\alpha$, we can still conclude that the sample and computational complexities for full recovery are $\Theta(K \operatorname{polylog}(n))$.

## III. Mixed-Coloring Algorithm for Noiseless Recovery

In this section, we provide details of the Mixed-Coloring algorithm in the noiseless setting. We first provide some primitives that serve as important ingredients in the algorithm, and then describe the design of query vectors and decoding algorithm in detail.

### A. Primitives

The algorithm makes uses of four basic primitives: **summation check**, **indexing**, **peeling**, and **guess-and-check**, which are described below.

**Summation Check:** Suppose that we generate two query vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ independently from some continuous distribution on $\mathbb{C}^n$, and a third query vector of the form $\boldsymbol{x}_1+\boldsymbol{x}_2$. Let $y_1$, $y_2$, and $y_3$ be the corresponding measurements. We check the sum of the measurements and in the noiseless case, if $y_3 = y_1+y_2$, then with probability one, we know that these three measurements are generated from the same parameter vector $\boldsymbol{\beta}^{(\ell)}$. In this case we call $\{y_1, y_2\}$ a *consistent pair* of measurements as they are from the same $\boldsymbol{\beta}^{(\ell)}$ (the third measurement $y_3$ is now redundant).

**Indexing:** The indexing procedure is to find the locations and values of the non-zero elements by carefully designed query vectors. In the noiseless case, this can be done by suitably designed *ratio test*. We sketch the idea of the ratio test here. Consider a consistent pair of measurements $\{y_1, y_2\}$ and corresponding query vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. We design the query vectors such that the information of the locations of the non-zero elements is encoded in the relative phase between $y_1$ and $y_2$. In particular, we generate $n$ i.i.d. random variables $r_j, j \in [n]$ uniformly distributed on the unit circle. Letting $W = e^{\mathbf{i}\frac{2\pi}{n}}$ where $\mathbf{i}$ is the imaginary unit, we set the $j$-th entries of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ to be either $x_{1,j} = x_{2,j} = 0$, or $x_{1,j} = r_j$ and $x_{2,j} = r_j W^{j-1}$. (The locations of the zeros are determined using sparse-graph codes and discussed later.) Below is an example of such a consistent pair of measurements and the corresponding linear system:

$$\begin{aligned}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{H}} \\ \boldsymbol{x}_2^{\mathrm{H}} \end{bmatrix} \boldsymbol{\beta}^{(1)} \\ &= \begin{bmatrix} 0 & r_2 & r_3 & 0 & 0 & r_6 & 0 & 0 \\ 0 & r_2 W & r_3 W^2 & 0 & 0 & r_6 W^5 & 0 & 0 \end{bmatrix} \boldsymbol{\beta}^{(1)}.\end{aligned} \tag{2}$$

Suppose that $\boldsymbol{\beta}^{(1)}$ is 3-sparse and of the form $\boldsymbol{\beta}^{(1)} = [0\ 0\ *\ 0\ *\ 0\ 0\ *]^{\mathrm{T}}$. There is only one non-zero element, $\beta_3^{(1)}$, that contributes to the measurements $y_1$ and $y_2$ . In this case the consistent measurement pair $\{y_1, y_2\}$ is called a *singleton*. A singleton can be detected by testing the integrality of the relative phase of the ratio $y_1/y_2$. In the above example, since $y_1 = r_3\beta_3^{(1)}$ and $y_2 = r_3 W^2 \beta_3^{(1)}$, we observe that $|y_1| = |y_2|$ and the relative phase $\angle(y_2/y_1) = 2 \cdot \frac{2\pi}{8}$ is an integral multiple of $\frac{2\pi}{8}$. We therefore know that with probability one, this consistent pair is a singleton, and moreover the corresponding non-zero element is located at the 3-rd coordinate with value $\beta_3^{(1)} = y_1/r_3$. We would like to remark that the indexing step can also be done using real-valued query vectors.

**Peeling:** The third ingredient of the decoder is peeling, i.e., iteratively reducing the problem by subtracting off recovered elements, in a Gaussian elimination-like manner. In the example above, suppose instead that $\boldsymbol{\beta}^{(1)}$ is 4-sparse, i.e., $\boldsymbol{\beta}^{(1)} = [0\ *\ *\ 0\ *\ 0\ 0\ *]^{\mathrm{T}}$, in which case the consistent pair

$$y_i = x_{i,2}\beta_2^{(1)} + x_{i,3}\beta_3^{(1)}, \quad i = 1, 2 \tag{3}$$

is associated with two non-zero elements of $\boldsymbol{\beta}^{(1)}$. If in a previous iteration of the algorithm we have recovered the location and value of $\beta_2^{(1)}$, then we can subtract/peel off this recovered element by $y_i \leftarrow y_i - x_{i,2}\beta_2^{(1)}$, for $i = 1, 2$.

The updated measurement pairs satisfy $y_i = x_{i,3}\beta_3^{(1)}, i = 1, 2$, and we have reduced the problem to a simpler form. In fact, in this case the pair $\{y_1, y_2\}$ becomes a singleton, to which the above ratio test can be applied to recover $\beta_3^{(1)}$.

**Guess-and-check:** The ratio test and peeling steps can be combined to detect that two non-zero elements are from the same parameter vectors. In the previous example (3), suppose instead that we recovered two elements $\beta_2^{(\ell_1)}$ and $\beta_3^{(\ell_2)}$ in previous iterations via ratio-testing another two consistent pairs that are singletons, but values of their labels $\ell_1$ and $\ell_2$ are unknown. We can still try to peel off $\beta_2^{(\ell_1)}$ from $\{y_1, y_2\}$; if the updated measurements $\{y_1, y_2\}$ pass the ratio test and recover a non-zero element with location 3 and value $\beta_3^{(\ell_2)}$, then we know that with probability one the non-zero elements $\beta_2^{(\ell_1)}$ and $\beta_3^{(\ell_2)}$ must come from the same parameter vector (the one that generates $\{y_1, y_2\}$), i.e., $\ell_1 = \ell_2 = 1$. In this case the peeling step is valid.

The continuing execution of these four primitives is made possible by the design of the query vectors using sparse-graph codes, which we describe next.

### B. Design of Query Vectors

As illustrated in Figure 3, we construct $M = \Theta(K)$ sets of query vectors (called *bins*). The query vectors in each bin are associated with some coordinates of the parameter vectors (i.e., the queries are non-zero only on those coordinates). The association between the coordinates and bins is determined by a $d$-left regular bipartite graph with $n$ left nodes (coordinates) and $M$ right nodes (bins), where each left node is connected to $d = \Theta(1)$ right nodes chosen independently uniformly at random. Each bin consists of three query vectors. The values of the non-zero elements of the first two query vectors are in the form of (2), enabling the ratio test. The third query vectors equals the sum of the first two and is used for the summation check.

If the query vectors in each bin were used only once, then we would have very few bins passing the summation check and hence few consistent pairs. Instead, we use the first two query vectors repeatedly for $R = \Theta(1)$ times, obtaining two

sets of measurements, each of size $R$ and called *type-I* and *type-II index measurements*. We use the third query vector $V = \Theta(1)$ times to obtain a set of *verification measurements*. We therefore have $2R + V$ measurements associated with each of the $M$ bins, hence a total of $m = (2R + V)M = \Theta(K)$ measurements, as shown in Figure 3. Using density evolution methods [12], we can find proper values of $d$, $R$, $V$, and $M$ such that successful recovery is guaranteed.

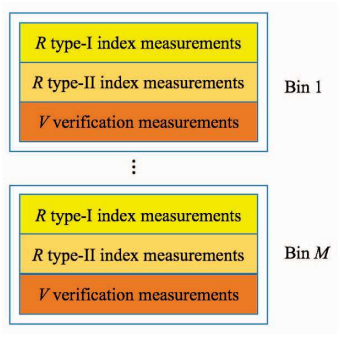



Fig. 3: $(2R + V)M$ query vectors.

### C. Decoding Algorithm

The decoding algorithm first finds consistent pairs (by summation check) in each bin, within which singletons are identified (by the ratio test). The ratio test also recovers the location and values of several non-zero elements, some of which can then be associated with the same $\boldsymbol{\beta}^{(\ell)}$ by guess-and-check. At this point, for each $\boldsymbol{\beta}^{(\ell)}$, we have recovered some of its non-zero elements (including their locations, values and labels). These steps are then repeated iteratively via peeling until no more non-zero elements can be found. Below we elaborate on these steps.

*a) Finding Consistent Pairs:* The decoding procedure starts by finding all the consistent pairs. In each bin, we perform summation checks on all triplets $(y_1, y_2, y_3)$ in which $y_1$, $y_2$, and $y_3$ are the type-I index measurement, type-II index measurement and verification measurement, respectively. If a triplet passes the summation check, then a consistent pair $\{y_1, y_2\}$ is found. Note that in each bin the number of triplets of the above form is a constant, so this step can be done in $\Theta(K)$ time. The subsequent steps of the algorithm are based on the consistent pairs found in this step.

*b) Recovering a Subset of Non-zero Elements:* Each non-zero element of the parameter vectors can be identified by its label-location-value triplet $(\ell, j, \beta_j^{(\ell)})$. We visualize these triplets (i.e., non-zero elements) as balls, as shown in Figure 1a, and initially their labels, locations and values are unknown. As before, a consistent pair associated with only one non-zero element is called a singleton, and we call this non-zero element a *singleton ball*. We run the ratio test on the consistent pairs to identify singletons and their associated singleton balls. The singleton balls found are illustrated in Figure 1b as shaded balls. The ratio test also recovers the locations and values of these singleton balls, although at this point we do not know the label $\ell$ of the balls.

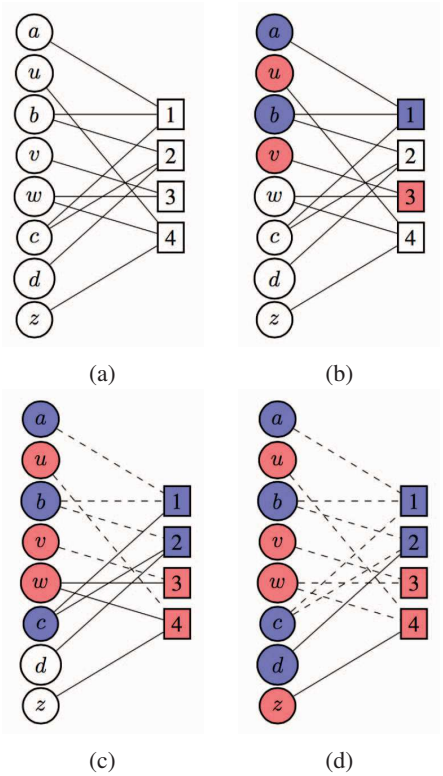



Fig. 4: Iterative decoding. If a ball is peeled off, the edges connected to it are shown in dashed lines. The colored balls in (b) are found by the giant component method. In (c) and (d), more balls are colored by iterative decoding.

The next step is crucial: For two singleton balls and a consistent measurement pair associated with the locations of these two balls, we run the guess-and-check operations to detect if these two singleton balls indeed have the same label (or equivalently, if the two non-zero elements are in the same parameter vector). If so, we connect these two balls with an edge, as shown in Figure 1b. Doing so creates a graph over the balls (i.e., non-zero elements), and each connected component of the graph is from a single parameter vector. Since each non-zero element is associated with a constant number of consistent pairs (due to using a $d$-left regular bipartite graph with constant $d$), this step can in fact be done efficiently in $\Theta(K)$ time without enumerating all the combinations of singleton ball pairs.

By carefully choosing the parameters $d$, $M$, $R$, and $V$, and using tools from random graph theory, we can ensure that with high probability the $L$ *largest* connected components (called *giant components*) correspond to the $L$ parameter vectors, and each of these components has size $\Theta(K)$. Then, the labels of the balls in these components are now identified. This is illustrated in Figure 1c for $L = 2$, where colors represent the labels. In summary, at this point we have recovered the labels, locations and values of a constant fraction of the non-zero elements (i.e., balls) of each parameter vector.

*c) Iterative Decoding:* The decoding procedure proceeds by identifying the labels of the remaining balls via iteratively applying the peeling and guess-and-check primi-

tives. The connected components in Figure 1c are therefore expanded, until no more changes can be made, as illustrated in Figure 1d.

We provide an example of this iterative procedure in Figure 4. Recall that the association between the coordinates of the parameter vectors and the bins (or consistent pairs) is determined by a bipartite graph. Here, we only show one consistent pair for each bin and omit the zero elements. The non-zero elements and the consistent pairs are shown as balls and squares, respectively, as in Figure 4a. The steps described in the last part recover a subset of these balls, which are shown in colors in Figure 4b. Now consider the measurement pair 1, which is associated with the balls $a$, $b$ and $c$. As $a$ and $b$ are recovered, we can peel them off from the measurement pair 1 to recover (by the ratio test) the label, location and value of the non-zero element represented by ball $c$. Similarly, peeling off the recovered ball $v$ from the measurement pair 3, recovers ball $w$, as illustrated in Figure 4c. We continue this process iteratively, peeling off balls recovered in the previous iterations to recover more balls. For example, we peel off the balls $b$ and $c$ from the measurement pair 2 to recover the ball $d$, and the ball $w$ from pair 4 to recover ball $z$, resulting in Figure 4d. So far we have described the Mixed-Coloring algorithm in the noiseless case.

## IV. Robust Mixed-Coloring Algorithm for Noisy Recovery

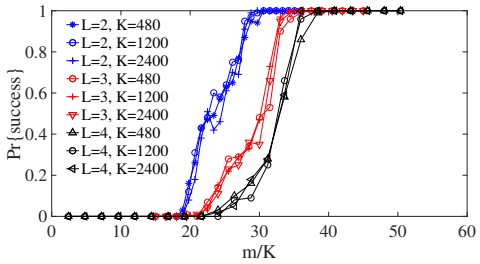
The overall structure of the Robust Mixed-Coloring algorithm is the same as its noiseless counterpart. In the presence of noise, the ratio test method for indexing and the summation check primitive need to be robustified, which are done by a modification of the query design. In particular, we design three types of query vectors. The first type, called *binary indexing* vectors, encodes the location information using binary representations with, $\lceil \log_2(n) \rceil$ bits (as opposed to using the relative phases in the noiseless case). A similar approach is considered in [13] for compressive phase retrieval. The second type is called *singleton verification* vectors, which are used for singleton detection. Using these two types of vectors we can modify the ratio test to achieve the same performance with noise. The third type of query vectors is used for *consecutive summation check*, which finds *consistent sets* of measurements.

In addition to the new query design, we also employ a noise reduction scheme. This is done by using each designed query vector (say $\boldsymbol{x}_i$) repeatedly for $R$ times and averaging the corresponding measurements from the same $\boldsymbol{\beta}^{(\ell)}$. In particular, these $R$ measurements are sampled i.i.d. from a mixture of two Gaussians with centers $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{(2)}$, so we use an EM algorithm initialized by moment methods to estimate the two centers. Using the result in [14], we prove that the EM-based noise reduction scheme succeeds under the conditions in Theorem 2, namely $R = \Theta(\mathrm{polylog}(n))$ and $\Delta/\sigma > \eta$. We refer the readers to Section **??** of the appendices for the details of the Robust Mixed-Coloring algorithm.

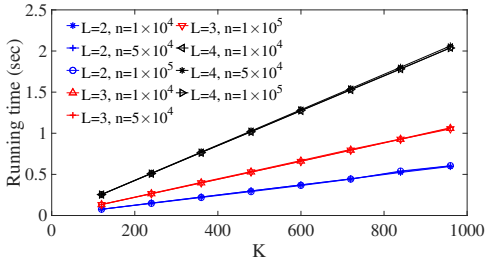
## V. Experimental Results

In this section, we test the sample and time complexities of the Mixed-Coloring algorithm in both noiseless and noisy cases to verify our theoretical results. We refer the readers to the appendices for more details of the experiments.

For the noiseless case, we use the optimal parameters $(d, R, V)$ from numerical calculations of the density evolution. For different values of $L, K, m$, we record the empirical success probability and running time averaged over 100 trials. Here, we use a sufficiently small $p^*$ so that the success event is equivalent to recovery of *all* the non-zero elements. The results are shown in Figure 5a. The phase transition occurs at some $C = m/K$ that matches the values in Table I predicted by our theory. Moreover, the running time is linear in $K$ and does not depend on $n$, as shown in Figure 5b.




(a) Probability of success $(n = 10^5)$



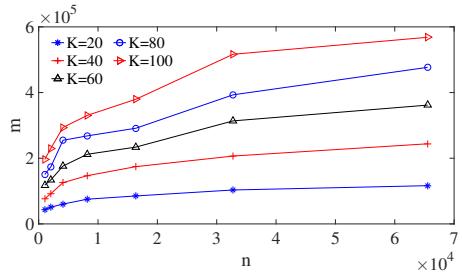

(b) Time complexity

Fig. 5: Success probability and running time in the noiseless case.

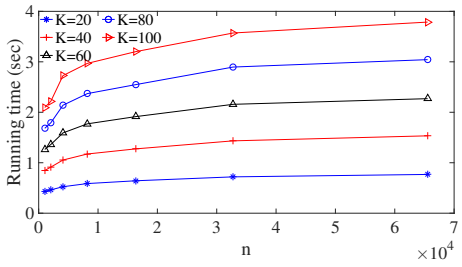
Similar experiments are performed for the noisy case using the Robust Mixed-Coloring algorithm, under the quantization assumption. Figure 6a shows the minimum number of queries $m$ required for 100 consecutive successes, for different $n$ and $K$. We observe that the sample complexity is linear in $K$ and sublinear in $n$. The running time exhibits a similar behavior, as shown in Figure 6b. Both observations agree with the prediction of our theory.

We also compare the Mixed-Coloring algorithm with a state-of-the-art EM-style algorithm (equivalent to alternating minimization in the noiseless setting) from [15]. These comparisons are not entirely fair, since our algorithm is based on carefully designed query vectors, while the algorithm in [15] uses random design, i.e., the entries of $\boldsymbol{x}_i$'s are i.i.d. Gaussian. However, this is exactly where the intellectual value of our work lies: we expose the gains available by careful design. We consider four test cases with $(L, n, K) = (2, 100, 20), (2, 500, 50), (2, 100, 100), (2, 500, 500),$ with

the first two cases being sparse problems and the last two being relatively dense problems. We find the minimum number of queries that leads to a 100% successful rate in 100 trials, and the average running time. As shown in Table II, in both sparse and dense problems, our Mixed-Coloring algorithm is several orders of magnitude faster. As for the sample complexity, our algorithm requires smaller number of samples in the sparse cases, while in dense problems, the sample complexity of our algorithm is within a constant factor (about 3) of that of the alternating minimization algorithm. For the noisy setting, our algorithm is most powerful in the high dimensional setting, i.e., large $n$, due to the polylog$(n)$ factors. However, in this setting, it takes extremely long time for the state-of-the-art algorithms such as [16] to converge, and thus, we do not present the comparison in the noisy setting.




(a) Sample complexity $(\Delta = 1, \sigma = 0.2)$




(b) Time complexity $(\Delta = 1, \sigma = 0.2)$

Fig. 6: Sample and time complexities of Robust Mixed-Coloring algorithm.

TABLE II: Comparison of two algorithms (M-C=Mixed-Coloring)

| $(n, K)$ | sample(M-C) / sample(EM) | speed(M-C) / speed(EM) |
|---|---|---|
| $(100, 20)$ | 0.57 | 124 |
| $(500, 50)$ | 0.33 | 368 |
| $(100, 100)$ | 2.78 | 19 |
| $(500, 500)$ | 3.00 | 37 |

We further test the Robust Mixed-Coloring algorithm when the quantization assumption is violated. For any $\beta \in \mathbb{R}$, we define $D(\beta) = \arg\min_{a \in \mathbb{D}} |a - \beta| \mathbf{1}(\beta \neq 0)$, where $\mathbf{1}(\cdot)$ denotes the indicator function. This means that $D(\beta)$ is the element in $\mathbb{D}$ which is the closest one to $\beta$, when $\beta \neq 0$. For a vector $\boldsymbol{\beta} \in \mathbb{R}^n$, we define $D(\boldsymbol{\beta}) = \{D(\beta_j)\}_{j=1}^n$. We define the *perturbation* of a vector $\boldsymbol{\beta}$ as Perturbation$(\boldsymbol{\beta}) = \max_{j \in [n]} |\beta_j - D(\beta_j)|/\Delta$.

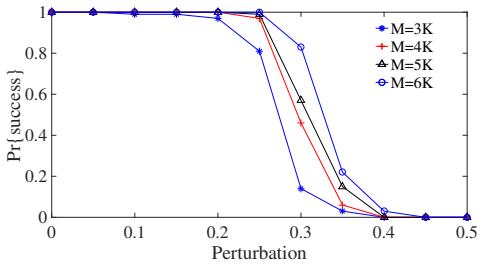



Fig. 7: Performance of Robust Mixed-Coloring algorithm with quantization assumption violated.

In this experiment, we generate sparse parameter vectors $\boldsymbol{\beta}^{(\ell)}$, $\ell \in [L]$ with a total number of $K$ non-zero elements. These non-zero elements are generated randomly while keeping the perturbation of the parameter vectors under a certain level by adding bounded noise to the quantized non-zero elements. We record the probability of success for different number of bins $M$ and different perturbation level. Here the success event is defined as recovery of $D(\boldsymbol{\beta}^{(\ell)})$ for all $\ell \in [L]$. The result is shown in Figure 7. We see that the Robust Mixed-Coloring algorithm works without the quantization assumption as long as the perturbations are not too large.

## VI. Related Work

### A. Mixtures of Regressions

Parameter estimation using the expectation-maximization (EM) algorithm is studied empirically in [17]. In [16], an $\ell_1$-penalized EM algorithm is proposed for the sparse setting. Theoretical analysis of the EM algorithm is difficult due to non-convexity. Progress was made in [15], [18] and [14] under stylized Gaussian settings with dense $\boldsymbol{\beta}$, for which a sample complexity of $\Theta(n \operatorname{polylog}(n))$ is proved given a suitable initialization of EM. The algorithm uses a grid search initialization step to guarantee that the EM algorithm can find the global optimal solution, with the assumption that the query vectors are i.i.d. Gaussian distributed. The computational complexity is polynomial of $n$. An alternative algorithm is proposed in [19], which achieves optimal $\mathcal{O}(n)$ sample complexity, but has high computational cost due to the use of semidefinite lifting. The algorithm in [20] makes use of tensor decomposing techniques, but suffers from a high sample complexity of $\mathcal{O}(n^6)$. In comparison, our approach has order optimal sample and time complexities by utilizing the potential design freedom. The classification version of this problem has also been studied in [21].

### B. Coding-theoretic Methods

Many modern error-correcting codes such as LDPC codes and polar codes [22] with their roots in communication problems, exploit redundancy to achieve robustness, and use structural design to allow for fast decoding. These properties of codes have recently found applications in statistical problems, including graph sketching [23], sparse covariance estimation [24], low-rank approximation [25], and discrete

inference [26]. Most related to our approach is the work in [13], [27], [28], which apply sparse graph codes with peeling-style decoding algorithms to compressive sensing and phase retrieval problems. In our setting we need to handle a mixture distribution, which requires more sophisticated query design and novel unmixing algorithms that go beyond the standard peeling-style decoding.

### C. Combinatorial and Dimension Reduction Techniques

Our results demonstrate the power of strategic query and coding theoretic tools in mixture problems, and can be considered as efficient linear sketching of a mixture of sparse vectors. In this sense, our work is in line with recent work that make uses of combinatorial and dimension reduction techniques in high-dimensional and large scale statistical problems. These techniques, such as locality-sensitive hashing [29], sketching of convex optimization [30], and coding-theoretic methods [31], allow one to design highly efficient and robust algorithms applicable to computationally challenging datasets without compromising statistical accuracy.

## VII. Conclusions

We propose the Mixed-Coloring algorithm as a query based learning algorithm for mixtures of sparse linear regressions. The design of the query vectors and the recovery algorithm are base sparse graph codes, and our scheme achieves order optimal sample and computational complexities in the noiseless case, and sublinear sample and time complexities in the presence of noise. Our experiments justified the theoretical results. In the noisy scenario, studying the Robust Mixed-Coloring algorithm with more than two parameter vectors and obtain theoretical results for the continuous alphabet can be two important future directions.

## References


[1] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," in *Computer Vision ECCV 2002*. Springer, 2002, pp. 543–560.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[3] A. Zhang, N. Fawaz, S. Ioannidis, and A. Montanari, "Guess who rated this movie: Identifying users through subspace clustering," *arXiv preprint arXiv:1208.1544*, 2012.

[4] R. De Veaux, "Mixtures of linear regressions," *Comp. Statistics & Data Analysis*, vol. 8, no. 3, 1989.

[5] E. Blackwell, C. F. M. de Leon, and G. E. Miller, "Applying mixed regression models to the analysis of repeated-measures data in psychosomatic medicine," *Psychosomatic Medicine*, vol. 68, no. 6, 2006.

[6] P. Deb and M. Holmes, "Estimates of use and costs of behavioural health care: a comparison of standard and finite mixture models," *Econometric Analysis of Health Data*, pp. 87–99, 2002.

[7] K. Viele and B. Tong, "Modeling with mixtures of linear regressions," *Statistics and Computing*, vol. 12, no. 4, pp. 315–330, 2002.

[8] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[9] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.

[10] R. Jansen, "A general mixture model for mapping quantitative trait loci by using molecular markers," *Theoretical and Applied Genetics*, vol. 85, no. 2-3, pp. 252–260, 1992.

[11] D. Yin, R. Pedarsani, X. Li, and K. Ramchandran, "Compressed sensing using sparse-graph codes for the continuous-alphabet setting," *54nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016.

[12] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.

[13] D. Yin, K. Lee, R. Pedarsani, and K. Ramchandran, "Fast and robust compressive phase retrieval with sparse-graph codes," in *IEEE International Symposium on Information Theory*, 2015, pp. 2583–2587.

[14] S. Balakrishnan, M. J. Wainwright, and B. Yu, "Statistical guarantees for the em algorithm: From population to sample-based analysis," *arXiv preprint:1408.2156*, 2014.

[15] X. Yi, C. Caramanis, and S. Sanghavi, "Alternating minimization for mixed linear regression," in *ICML*, 2014, pp. 613–621.

[16] N. Städler, P. Bühlmann, and S. Van De Geer, "$\ell_1$-penalization for mixture regression models," *Test*, vol. 19, no. 2, pp. 209–256, 2010.

[17] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation*, vol. 80, no. 2, pp. 201–225, 2010.

[18] X. Yi, C. Caramanis, and S. Sanghavi, "Solving a mixture of many random linear equations by tensor decomposition and alternating minimization," *arXiv preprint arXiv:1608.05749*, 2016.

[19] Y. Chen, X. Yi, and C. Caramanis, "A convex formulation for mixed regression with two components: Minimax optimal rates," *arXiv preprint arXiv:1312.7006*, 2013.

[20] A. T. Chaganty and P. Liang, "Spectral experts for estimating mixtures of linear regressions," in *ICML*, 2013, pp. 1040–1048.

[21] Y. Sun, S. Ioannidis, and A. Montanari, "Learning mixtures of linear classifiers." in *ICML*, 2014, pp. 721–729.

[22] E. Arikan, "Channel polarization a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory,*, vol. 55, no. 7, 2009.

[23] X. Li and K. Ramchandran, "An active learning framework using sparse-graph codes for sparse polynomials and graph sketching," in *NIPS*, 2015, pp. 2161–2169.

[24] R. Pedarsani, K. Lee, and K. Ramchandran, "Sparse covariance estimation based on sparse-graph codes," in *Annual Allerton Conference on Communication, Control, and Computing*, 2015.

[25] S. Ubaru, A. Mazumdar, and Y. Saad, "Low rank approximation using error correcting coding matrices," in *ICML*, 2015, pp. 702–710.

[26] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman, "Low-density parity constraints for hashing-based discrete integration," in *ICML*, 2014, pp. 271–279.

[27] X. Li, S. Pawar, and K. Ramchandran, "Sub-linear time support recovery for compressed sensing using sparse-graph codes," *arXiv preprint arXiv:1412.7646*, 2014.

[28] R. Pedarsani, D. Yin, K. Lee, and K. Ramchandran, "Phasecode: Fast and efficient compressive phase retrieval based on sparse-graph codes," *IEEE Transactions on Information Theory*, 2017.

[29] I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Nearest neighbor based greedy coordinate descent," in *NIPS*, 2011, pp. 2160–2168.

[30] M. Pilanci and M. J. Wainwright, "Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares," *arXiv preprint arXiv:1411.0347*, 2014.

[31] D. Achlioptas and P. Jiang, "Stochastic integration via error-correcting codes," in *UAI*, 2015.