2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Towards Learning Sparsely Used Dictionaries with Arbitrary Supports

Pranjal Awasthi
Department of Computer Science
Rutgers University
Email: pranjal.awasthi@rutgers.edu

Aravindan Vijayaraghavan
Department of Electrical Engineering and Computer Science
Northwestern University
Email: aravindv@northwestern.edu

Abstract*—Dictionary learning is a popular approach for inferring a hidden basis in which data has a sparse representation. There is a hidden dictionary or basis A which is an $n \times m$ matrix, with $m > n$ typically (this is called the over-complete setting). Data generated from the dictionary is given by $Y = AX$ where X is a matrix whose columns have supports chosen from a distribution over k-sparse vectors, and the non-zero values chosen from a symmetric distribution. Given Y, the goal is to recover A and X in polynomial time (in m, n). Existing algorithms give polynomial time guarantees for recovering incoherent dictionaries, under strong distributional assumptions both on the supports of the columns of X, and on the values of the non-zero entries. In this work, we study the following question: *can we design efficient algorithms for recovering dictionaries when the supports of the columns of X are arbitrary?

To address this question while circumventing the issue of non-identifiability, we study a natural semirandom model for dictionary learning. In this model, there are a large number of samples $y = Ax$ with arbitrary k-sparse supports for x, along with a few samples where the sparse supports are chosen uniformly at random. While the presence of a few samples with random supports ensures identifiability, the support distribution can look almost arbitrary in aggregate. Hence, existing algorithmic techniques seem to break down as they make strong assumptions on the supports.

Our main contribution is a new polynomial time algorithm for learning incoherent over-complete dictionaries that provably works under the semirandom model. Additionally the same algorithm provides polynomial time guarantees in new parameter regimes when the supports are fully random. Finally, as a by product of our techniques, we also identify a minimal set of conditions on the supports under which the dictionary can be (information theoretically) recovered from polynomially many samples for almost linear sparsity, i.e., $k = \widetilde{O}(n)$.

***Keywords*-beyond worst-case analysis; semi-random models; dictionary learning**

I. Introduction

In many machine learning applications, the first step towards understanding the structure of naturally occurring data such as images and speech signals is to find an appropriate basis in which the data is sparse. Such sparse representations lead to statistical efficiency and can often uncover semantic features associated with the data. For example images are often represented using the *SIFT* basis [1]. Instead of designing an appropriate basis by hand, the goal of dictionary learning is to algorithmically learn from data, the basis (also known as the dictionary) along with the data's sparse representation in the dictionary. This problem of dictionary learning or sparse coding was first formalized in the seminal work of Olshausen and Field [2], and has now become an integral approach in unsupervised learning for feature extraction and data modeling.

The dictionary learning problem is to learn the unknown dictionary $A \in \mathbb{R}^{n \times m}$ and recover the sparse representation X given data Y that is generated as follows. The typical setting is the "over-complete" setting when $m > n$. Each column A_i of A is a vector in $\mathbb{R}^n$ and is part of the over-complete basis. Data is then generated by taking random sparse linear combinations of the columns of A. Hence the data matrix $Y \in \mathbb{R}^{n \times N}$ is generated as $Y = AX$, where $X \in \mathbb{R}^{m \times N}$ captures the representation of each of the N data points[1]. Each column of X is a vector drawn from a distribution $\mathcal{D}^{(s)} \odot \mathcal{D}^{(v)}$. Here $\mathcal{D}^{(s)}$ is a distribution over k sparse vectors in $\{0,1\}^m$ and represents the *support distribution*. Conditioning on support of the column x, each non-zero value is drawn independently from $\mathcal{D}^{(v)}$, which represents the *value distribution*.

The goal of recovering (A, X) from Y is particularly challenging in the over-complete setting – notice that even if A is given, finding the matrix X with sparse supports such that $Y = AX$ is the sparse recovery or compressed sensing problem which is NP-hard in general [3]. A beautiful line of work [4], [5], [6], [7] gives polynomial time recovery of X (given A) under certain assumptions about A like Restricted Isometry Property (RIP) and incoherence. See Section II for formal definitions.

While there have been several heuristics and algorithms proposed for dictionary learning, the first rigorous polynomial time guarantees were given by Spielman et al. [8] who focused on the *full rank* case, i.e., $m = n$. They assumed that the support distribution $\mathcal{D}^{(s)}$ is uniformly random (each entry is non-zero independently with probability $p = k/m = 1/\sqrt{m}$) and the value distribution $\mathcal{D}^{(v)}$ is a symmetric sub-Gaussian distribution, and this has subsequently been improved by [9] to handle almost linear sparsity. The first algorithmic gua-

[1]In general there can also be noise in the model where each column of Y is given by $y = Ax + \psi$ where ψ is a noise vector of small norm. In this paper we focus on the noiseless case, though our algorithms are also robust to inverse polynomial error in each sample.

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00035

rantees for learning over-complete dictionaries (m can be larger than n) from polynomially many (in m, n) samples, and in polynomial time were independently given by Arora et al. [10] and Agarwal et al. [11].In particular, the work of [12] and its follow up work [12] provide guarantees for sparsity up to $n^{1/2}/\log m$, and also assumes slightly weaker assumptions on the support distribution $\mathcal{D}^{(s)}$, requiring it to be approximately $O(1)$-wise independent. The works of [13] and [14] gives Sum of Squares (SoS) based quasi-polynomial time algorithms (and polynomial time guarantees in some settings) to handle almost linear sparsity under similar distributional assumptions. See Section I-C for a more detailed discussion and comparison of these works.

While these algorithms give polynomial time guarantees even in over-complete settings, they crucially rely on strong distributional assumptions on both the support distribution $\mathcal{D}^{(s)}$ and the value distribution $\mathcal{D}^{(v)}$. Curiously, it is not known whether these strong assumptions are necessary to recover A, X from polynomially many samples, even from an information theoretic point of view. This motivates the following question that we study in this work:

Can we design efficient algorithms for learning over-complete dictionaries when the support distribution is essentially arbitrary?

As one might guess, the above question as stated, is ill posed since recovering the dictionary is impossible if there is a column that is involved in very few samples[2]. In fact we do not have a good understanding of when there is a unique (A, X) pair that explains the data (this is related to the question of identifiability of the model). However, consider the following thought-experiment: suppose we have an instance with a large number of samples, each of the form $y = Ax$ with x being an arbitrary sparse vector. In addition, suppose we have a few samples (N_0 of them) that are drawn from the standard dictionary learning model where the supports are random. The mere presence of the samples with random supports will ensure that there is a unique dictionary A that is consistent with *all* the samples (as long as $N_0 = \Omega(n^2)$ for example). On the other hand, since most of the samples have arbitrary sparse supports, the aggregate distribution looks fairly arbitrary[3]. This motivates a natural semirandom model towards understanding dictionary learning when the sparse supports are arbitrary.

The semirandom model: In this model we have N samples of the form $y = Ax$ with most of them having arbitrary k-sparse supports for x, and a few samples (N_0 of them) that are drawn from the random model for dictionary learning. We will use $\widetilde{\mathcal{D}}^{(s)}$ to represent the arbitrary distribution over k-sparse supports and $\mathcal{D}_R^{(s)}$ to represent the random distribution over k-sparse supports (as considered in prior works) and a parameter β to represent the fraction of samples from $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$ (it will be instructive to think of β as very small e.g., an inverse polynomial in n, m). N samples from the semirandom model $\mathcal{M}_\beta(\mathcal{D}_R^{(s)}, \widetilde{\mathcal{D}}^{(s)}, \mathcal{D}^{(v)})$ are generated as follows.

1) The supports of $N_0 = \beta N$ samples $x^{(1)}, \ldots, x^{(N_0)}$ are generated from the random distribution $\mathcal{D}_R^{(s)}$ over k-sparse $\{0,1\}^m$ vectors [4].
2) The adversary chooses the k-sparse supports of $N_1 = (1-\beta)N$ samples arbitrarily (or equivalently from an arbitrary distribution $\widetilde{\mathcal{D}}^{(s)}$). Note that the adversary can also see the supports of the N_0 "random" samples.
3) The values of each of the non-zeros in $X = \{x^{(\ell)} : \ell \in [N]\}$ are picked independently from the value distribution $\mathcal{D}^{(v)}$ e.g., a Rademacher distribution (± 1 with equal probability).
4) The $x^{(1)}, \ldots, x^{(N)}$ are reordered randomly to form matrix $X \in \mathbb{R}^{m \times N}$ and the data matrix $Y = AX$. Y is the instance of the dictionary learning problem.

The samples that are generated in step 1 will be referred to as the random portion (or random samples), and the samples generated in step 2 will be referred to adversarial samples. As mentioned earlier, the presence of just the random portion ensures that the model is identifiable (assuming $\beta N = n^{\Omega(1)}$) from known results, and there is unique solution A. The additional samples that are added in step 2 represent more k-sparse combinations of the columns of A – hence, intuitively the adversary is only helpful by presenting more information about A (such adversaries are often called monotone adversaries). On the other hand, the fraction of random samples β can be very small (think of $\beta = O(1/\mathrm{poly}(n))$) – hence the adversarial portion of the data can completely overwhelm the random portion. Further, the support distribution $\widetilde{\mathcal{D}}^{(s)}$ chosen by the adversary (or the supports of the adversarial samples) could have arbitrary correlations and also depend on the the support patterns in the random portion. Hence, the support distribution can look very adversarial, and this is challenging for existing algorithmic techniques, which seem to break down in this setting (see Sections I-C and I-B).

Semirandom models starting with works of [15], [16] have been a very fruitful paradigm for interpolating between average-case analysis and worst-case analysis. Further, we believe that studying such semirandom models for unsupervised learning problems will be very effective in identifying robust algorithms that do not use strong distributional properties of the instance. For instance, algorithms based on convex relaxations for related problems like compressed sensing [6] and matrix completion [17] are robust in the presence of a similar monotone adversary where there are additional

[2]See the full version for a more interesting example.

[3]since we do not know which of the samples are drawn with random support.

[4]More generally, $\mathcal{D}_R^{(s)}$ can be any distribution that is τ-negatively correlated – here $\forall S$ s.t. $|S| = O(\log m), i \notin S$, the probability $\mathbb{P}[i \in \mathrm{supp}(x) \mid S \subset \mathrm{supp}(x)] \leq \tau k/m$, and $\mathbb{P}[i \in supp(x)] \approx k/m$.

arbitrary observations in addition to the random observations.

A. Our Results

We present a new polynomial time algorithm for dictionary learning that works in the semirandom model and obtain new identifiability results under minimal assumptions about the sparse supports of X. We give an overview of our results for the simplest case, when the value distribution $\mathcal{D}^{(v)}$ is a Rademacher distribution i.e., each non-zero value x_i is either $\{+1, -1\}$ with equal probability. These results also extend to a more general setting where the value distribution $\mathcal{D}^{(v)}$ can be a mean-zero symmetric distribution supported in $[-C, -1] \cup [1, C]$ for a constant $C > 1$ – this is called *Spike-and-Slab* model [18] and has been considered in past works on sparse coding [10]. As with existing results on recovering dictionaries in the over-complete setting, we need to assume that the matrix satisfies some incoherence or Restricted Isometry Property (RIP) conditions (these are standard assumptions even in the sparse recovery problem when A is given). A matrix A is (k, δ)-RIP iff $(1-\delta)\|x\|_2 \leq \|Ax\|_2 \leq (1+\delta)\|x\|_2$ for all k-sparse vectors, and a matrix is μ-incoherent iff $|\langle A_i, A_j\rangle| \leq \mu/\sqrt{n}$ for every two columns $i \neq j \in [m]$. Random $n \times m$ matrices satisfy the (k, δ)-RIP property as long as $k = O(\delta n/\log(\frac{n}{\delta k}))$ [19], and are $\mu = O(\sqrt{\log m})$ incoherent. Please see Section II for the formal model and assumptions.

Our main result is a polynomial time algorithm for learning over-complete dictionaries when we are given samples from the semirandom model proposed above.

Informal Theorem I.1 (Polytime algorithm for semirandom model)**.** *Consider a dictionary $A \in \mathbb{R}^{n\times m}$ that is μ-incoherent with spectral norm σ. There is a polynomial time algorithm that given $\mathrm{poly}(n, m, k, 1/\beta)$ samples generated from the semirandom model (with β fraction random samples) with sparsity $k \leq \sqrt{n}/(\mu^{O(1)}(\sigma m/n)^{O(1)}\mathit{polylog} m)$, recovers with high probability the dictionary A up to arbitrary (inverse-polynomial) accuracy (up to relabeling the columns, and scaling by ± 1)*[5].

Please see Theorem V.1 for a formal statement. The above algorithm recovers the dictionary up to arbitrary accuracy in the semirandom model for sparsity $k = \widetilde{O}(n^{1/2})$ – as we will see soon, this is comparable to the state-of-the-art polynomial time guarantees even when there are no adversarial samples. By using standard results from sparse recovery [6], [7], one can then use our knowledge of A to recover X. We emphasize in the above bounds that the sparsity assumption and recovery error do not have any dependence on β the fraction of samples generated from the random portion. The dependence on $1/\beta$ in the sample complexity simply ensures that there are a few samples from the random portion in the generated data.

When there are no additional samples from the adversary i.e., $\beta = 1$, our algorithm in fact handles a significantly larger sparsity of $k = \widetilde{O}(m^{2/3})$

Informal Theorem I.2 (Beyond $\sqrt{n}$ with no adversarial supports ($\beta = 1$))**.** *Consider a dictionary $A \in \mathbb{R}^{n\times m}$ that is μ-incoherent and $(k, 1/\mathit{polylog} m)$-RIP with spectral norm σ. There is a polynomial time algorithm that given $\mathrm{poly}(n, m, k)$ samples generated from the "random" model with sparsity $k \leq n^{2/3}/(\mu^{O(1)}(\sigma m/n)^{O(1)}\mathit{polylog} m)$, recovers with high probability the dictionary A up to arbitrary accuracy.*

Please see Theorem VI.1 for a formal statement. For the sake of comparison, consider the case when the amount of over-completeness is $\widetilde{O}(1)$ or even n^{ε} for some small constant $\varepsilon > 0$ i.e., $m/n, \sigma \leq n^{\varepsilon}$.[6] The results of Arora et al. [10], [12] recover the dictionaries for sparsity $k = \widetilde{O}(\sqrt{n})$, when there are no adversarial samples. On the other hand, sophisticated algorithms based on Sum-of-Squares (SoS) relaxations [13], [14] give quasi-polynomial time guarantees in general (and polynomial time guarantees when $\sigma = O(1)$) for sparsity going up to $k = O(m/\mathrm{polylog} m)$ when there are no adversarial samples. Hence, our algorithm gives polynomial time guarantees in new settings when sparsity $k = \omega(\sqrt{n})$ even in the absence of any adversarial samples (Theorem I.2), and at the same time gives polynomial time guarantees for $k = \widetilde{O}(\sqrt{n})$ in the semirandom model even when the supports are almost arbitrary. Please see Section I-C for a more detailed comparison.

A key component of our algorithm that is crucial in handling the semirandom model is a new efficient procedure that allows us to test whether a given unit vector is close to a column of the dictionary A. In fact this procedure works up to sparsity $k = O(n/\mathrm{polylog}(m))$.

Informal Theorem I.3 (Test Candidate Column)**.** *Given any unit vector $z \in \mathbb{R}^n$, there is a polynomial time algorithm (Algorithm 1) that uses $\mathrm{poly}(m, n, k, 1/\eta_0)$ samples from the semirandom model with the sparsity $k \leq n/\mathit{polylog}(m)$ and the dictionary A satisfying $(k, \delta = 1/\mathit{polylog}(m))$-RIP property, that with probability at least $1 - \exp(-n^2)$:*

- (Completeness) *Accepts z if $\exists i \in [m],\ b \in \{\pm 1\}$ s.t. $\|z - bA_i\|_2 \leq 1/\mathit{polylog}(m)$.*
- (Soundness) *Rejects z if $\|z - bA_i\|_2 > 1/\mathrm{poly}\log(m)$ for every $i \in [m],\ b \in \{\pm 1\}$.*

Moreover in the first case, the algorithm also returns a vector $\widehat{z}$ s.t. $\|\widehat{z} - bA_i\|_2 \leq \eta_0$, where η_0 represents the desired inverse polynomial accuracy.

[5]We will recover a dictionary $\widehat{A}$ such that $\|\widehat{A}_i - b_i A_i\|_2 \leq \eta_0$ for some $b \in \{-1, 1\}^m$, where η_0 is the desired inverse-polynomial accuracy. While we state our guarantees for the noiseless case of $Y = AX$, our algorithms are robust to inverse polynomial additive noise.

[6]The parameter σ is an analytic measure of over-completeness; for any dictionary A of size $n \times m$, $\sigma \geq \sqrt{m/n}$. Conversely, one can also upper bound σ in terms of m/n under RIP-style assumptions. When the columns of A are random, then $\sigma = O(\sqrt{m/n})$; otherwise, $\sigma = O(\sqrt{m/k})$ when A is $(k, O(1))$-RIP.

Please see Theorem III.1 for a formal statement[7]. Our test is very simple and proceeds by computing inner products of the candidate vector z with samples and looking at the histogram of the values. Nonetheless, this provides a very powerful subroutine to discard vectors that are not close to any column. The full algorithm then proceeds by efficiently finding a set of candidate vectors (by simply considering appropriately weighted averages of all the samples), and running the testing procedure on each of these candidates. The analysis of the candidate-producing algorithm requires several ideas such as proving new concentration bounds for polynomials of rarely occurring random variables, which we describe in Section I-B.

In fact, the above test procedure works under more general conditions about the support distribution. This immediately implies *polynomial identifiability* for near-linear sparsity $k = O(n/\text{polylog}m)$, by simply applying the procedure to every unit vector in an appropriately chose ε-net of the unit sphere.

Informal Theorem I.4 (Polynomial Identifiability for Rademacher Value Distribution)**.** *Consider a dictionary $A \in \mathbb{R}^{n\times m}$ that is $(k, \delta = 1/polylog(m))$-RIP property for sparsity $k \leq n/polylog(m)$ and suppose we are given $N = \text{poly}(n, m, k, 1/\beta)$ samples with arbitrary k-sparse supports that satisfies the following condition:*

$\forall i_1, i_2, i_3 \in [m]$, there at least a few samples (at least $1/\text{poly}(n)$ fraction) $y = Ax$ such that $i_1, i_2, i_3 \in supp(x)$.

Then, there is a algorithm (potentially exponential runtime) that recovers with high probability a dictionary $\widehat{A}$ such that $\|\widehat{A}_i - b_i A_i\|_2 \leq 1/\text{poly}(m)$ for some $b \in \{-1, 1\}^m$ (up to relabeling the columns).

Please see Corollary IV.2 for a formal statement, and Corollary III.3 for related polynomial identifiability results under more general value distributions.

The above theorem proves polynomial identifiability for arbitrary set of supports as long as every triple of columns i_1, i_2, i_3 co-occur i.e., there are at least a few samples where they jointly occur (this would certainly be true if the support distribution has approximate three-wise independence). On the other hand in the full version of the paper, we complement this by proving a *non-identifiability* result using an instance that does not satisfy the "triples" condition, but where every pair of columns co-occur. Hence, Corollary IV.2 gives polynomial identifiability under arguably minimal assumptions on the supports. To the best of our knowledge, prior identifiability results were only known through the algorithmic results mentioned above, or using $n^{O(k)}$ many samples. Hence, while designed with the semirandom model in mind, our test procedure also allows us to shed new light on the information-theoretic problem of polynomial identifiability with adversarial supports.

[7]The above procedure is also noise tolerant – it is robust to adversarial noise of $1/\text{polylog}(n)$ in each sample.

Developing polynomial time algorithms that handle a sparsity of $k = \widetilde{O}(n)$ under the above conditions (e.g., Theorem I.4) that guarantee polynomial identifiability, or in the semirandom model, are interesting open questions.

B. Technical Overview

We now give an overview of the technical ideas involved in proving our algorithmic and identifiability results. Some of these ideas are also crucial in handling sparsity of $k = \omega(\sqrt{n})$ in the random model. Further, as we will see in the discussion that follows, the algorithm will make use of samples from both the random portion and the semirandom portion for recovering the columns. For the sake of exposition, let us restrict our attention to the value distribution being Rademacher i.e., each non-zero x_i is either $+1$ or -1 independently with equal probability.

Challenges with semirandom model for existing approaches: We first describe the challenges and issues that come in the semirandom model, and more generally when dealing with arbitrary support distributions. Many algorithms for learning over-complete dictionaries typically proceed by computing aggregate statistics of the samples e.g., appropriate moments of the samples $y = Ax$ (where $x \sim \mathcal{D}$), and then extracting individual columns of the dictionary – either using spectral approaches [12] or using tensor decompositions [13], [14]. However, in the semirandom model, the adversary can generate many more samples with adversarial supports, and dominate the number of random samples (it can be $\text{poly}(n)$ factor larger) — this can completely overwhelm the contribution of the random samples to the aggregate statistic . In fact, the supports of these adversarial samples can depend on the random samples as well.

To further illustrate the above point, let us consider the algorithm of Arora et al. [12]. They guess two fixed samples $u^{(1)} = A\zeta^{(1)}, u^{(2)} = A\zeta^{(2)}$ and consider the statistic

$$\begin{aligned} B &= \mathop{\mathbb{E}}_{y=Ax}\Big[\langle y, u^{(1)}\rangle\langle y, u^{(2)}\rangle y \otimes y\Big] \\ &= \sum_{i\in[m]} \big(\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\big[x_i^4\big]\langle A_i, \zeta^{(1)}\rangle\langle A_i, \zeta^{(2)}\rangle\big)\cdot A_i \otimes A_i + \\ &\quad + \sum_{i\neq i'} \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[x_i^2 x_{i'}^2]\big(\langle A_i, \zeta^{(1)}\rangle\langle A_{i'}, \zeta^{(2)}\rangle A_i \otimes A_{i'} + \dots\big) \end{aligned} \tag{1}$$

To recover the columns of A there are two main arguments involved. For the correct guess of $u^{(1)}, u^{(2)}$ with $\text{supp}(\zeta^{(1)}), \text{supp}(\zeta^{(2)})$ containing exactly one co-ordinate in common e.g., $i = 1$, they show that one gets $B = q_1 A_1 A_1^T + E$ where $\|E\| = o(q_1)$. In this way A_1 can be recovered up to reasonable accuracy (akin to *completeness*). To argue that $\|E\| = o(q_1)$, one can use the randomness in the support distribution to get that $\mathbb{E}[x_i^2 x_{i'}^2] = O(k^2/m^2)$ is significantly smaller (by a factor of approximately k/m) compared to $\mathbb{E}[x_1^2] \approx k/m$. On the other hand, one also needs to argue that for the wrong guess of $u^{(1)}, u^{(2)}$, the

resulting matrix B is not close to rank 1 (*soundness*). The argument here, again relies crucially on the randomness in the support distribution.

In the semirandom model, both completeness and soundness arguments are affected by the power of the adversary. For instance, if the adversary generates samples such that a subset of co-ordinates $T \subseteq [m]$ co-occur most of the time, then for every $i, i' \in T$, $\mathbb{E}[x_i^2 x_{i'}^2] = \Omega(\mathbb{E}[x_i^2])$. Hence, completeness becomes harder to argue since the cross-terms in (1) can be much larger (particularly for $k = \Omega(m^{1/8})$). The more critical issue is with soundness, since it is very hard to control and argue about the matrices B that are produced by incorrect guesses of $u^{(1)}, u^{(2)}$ (note that they can also be from the portion with adversarial support). For the above strategy in particular, there are adversarial supports and choices of samples such that B is close to rank 1 but whose principal component is not aligned along any of the columns of A (e.g., it could be along $\sum_{i \in T} A_i$). We now discuss how we overcome these challenges in the semirandom model.

Testing for a Single Column of the Dictionary: A key component of our algorithm is a new efficient procedure, which when given a candidate unit vector z tests whether z is indeed close to one of the columns of A (up to signs) or is far from every column of the dictionary i.e., $\|z - bA_i\|_2 > \eta$ for every $i \in [m], b \in \{-1, 1\}$ (η can be chosen to be $1/\mathrm{poly}\log(n)$ and the accuracy can be amplified later). Such a procedure can be used as a sub-routine with any algorithm in the semirandom model since it addresses the challenge of ensuring soundness. We can feed a candidate set of test vectors generated by the algorithm and discard the spurious ones.

The test procedure (Algorithm 1) is based on the following observation: if $z = bA_i$ for some column $i \in [m]$ and $b \in \{\pm 1\}$, then the distribution of $|\langle z, Ax\rangle|$ will be bimodal, depending on whether x_i is non-zero or not. This is because

$$|\langle bA_i, Ax\rangle| = |x_i| \pm \Big|\sum_{j \neq i} \langle A_i, A_j\rangle x_j\Big| = |x_i| \pm o(1) \quad \text{w.h.p.},$$

when A satisfies the RIP property (or incoherence). Hence Algorithm TESTCOLUMN (Algorithm 1) just computes the inner products $|\langle z, Ax\rangle|$ with polynomially many samples (it could be from the random or adversarial portion), and checks if they are always close to 0 or 1, with a non-negligible fraction of them (roughly k/m fraction, if each of the i occur equally often) taking a value close to 1.

The challenge in proving the correctness of this test is the soundness analysis: if unit vector z is far from any column of A_i, then we want to show that the test fails with high probability. Consider a candidate z that passes the test, and let $\alpha_i := \langle z, A_i\rangle$. Suppose $|\alpha_i| = o(1)$ for each $i \in [m]$ (so it is far from every column). For a sample $y = Ax$ with $\mathrm{supp}(x) = S$,

$$\langle z, Ax\rangle = \sum_{i \in S} x_i \langle z, A_i\rangle = \sum_{i \in S} \alpha_i x_i. \tag{2}$$

The quantity $\langle z, Ax\rangle$ is a weighted sum of symmetric, independent random variables x_i, and the variance of $\langle z, Ax\rangle$ equals $\|\alpha_S\|_2^2 = \sum_{i \in S} \alpha_i^2$. When $\|\alpha_S\|_2 = \Omega(1)$, Central Limit Theorems like the Berry-Esséen theorem tells us that the distribution of the values of $\langle z, Ax\rangle$ is close to a Normal distribution with $\Omega(1)$ variance. In this case, we can use anti-concentration of a Gaussian to prove that $|\langle z, Ax\rangle|$ takes a value bounded away from 0 or 1 (e.g., in the interval $[\frac{1}{4}, \frac{3}{4}]$) with constant probability. However, the variance $\|\alpha_S\|_2^2 = \sum_{i \in S} \alpha_i^2$ can be much smaller than 1 (for a random unit vector z, we expect $\|\alpha_S\|_2^2 = O(k/n)$). In general, we have very little control over the α_S vector since the candidate z is arbitrary. For an arbitrary spurious vector z, we need to argue that either $|\langle z, Ax\rangle|$ (almost) never takes large values close to 1, or takes values bounded away from 0 and 1 (e.g., in $[0.1, 0.9]$) for a non-negligible fraction of the samples.

The correctness of our test relies crucially on such an anti-concentration statement (see the full version of the paper), which may be of independent interest. Roughly speaking, we prove using a careful coupling argument that when random variable Z is a weighted sum of independent random variables as in (2), if $|Z|$ takes values close to t with non-negligible probability κ, then $|Z|$ also takes values in $[t/6, t/2]$ with probability at least $\Omega(\kappa)$ (the constants e.g., $1/6$ can be picked more generally).

The above test works for $k = O(n/\mathrm{poly}\log(m))$, only uses the randomness in the non-zero values, and works as long as the co-efficients $|\alpha_i|$ are all small compared to $t = 1$ i.e., $\|\alpha_S\|_\infty < ct$.[8] The full proof of the test uses a case analysis depending on whether there are some large co-efficients, and uses such a large coefficient in certain "nice" samples that exist under mild assumptions (e.g., in a semirandom model), along with the aforementioned lemma to prove that a unit vector z that is far from every column fails the test with high probability (the failure probability can be made to be $\exp(-n^2)$). Given the test procedure, to recover the columns of the dictionary, it now suffices (because of Algorithm 1) to design an algorithm that produces a set of candidate unit vectors that includes the columns of A.

Identifiability: The test procedure immediately implies polynomial identifiability (i.e., with polynomially many samples) for settings where the test procedure works, by simply running the test procedure on every unit vector in an ε-net of the unit sphere. When the value distribution $\mathcal{D}^{(v)}$ is a

[8]There are certain configurations where $|\alpha_i|$ are large, for which the above anti-concentration statement is not true. For example when $\alpha_1 = \alpha_2 = 1/2$ and 0 for rest of $i \in S$, then any ± 1 combination of α_1, α_2 is in $\{-1, 0, 1\}$. In fact, the non-identifiable instances have bad candidates z which precisely result in such combinations. However, we prove that this is essentially the only bad situation for this test.

Rademacher distribution, we prove that just a condition on the co-occurrence of every triple suffices to construct such a test procedure. This implies the polynomial identifiability results of Theorem I.4 for sparsity up to $k = O(n/\text{polylog}n)$. For more general value distributions defined in Section II, it just needs to hold that that there are a few samples where a given column i appears, but a few other $O(\log n)$ given columns do not appear (e.g., for example when a subset of samples satisfy very weak pairwise independence). This condition suffices for Algorithm 1 to work for sparsity $k = O(n/\text{polylog}n)$– and implies the identifiability results in Corollary III.3.

Efficiently Producing Candidate Vectors: Our algorithm for producing candidate vectors is inspired by the initialization algorithm of [12]. We guess $2L-1$ samples $u^{(1)} = A\zeta^{(1)}, u^{(2)} = A\zeta^{(2)}, \ldots, u^{(2L-1)} = A\zeta^{(2L-1)}$ for some appropriate constant L, and simply consider the weighted average of all samples given by

$$v = \mathop{\mathbb{E}}_{y}\Big[\langle y, A\zeta^{(1)}\rangle\langle y, A\zeta^{(2)}\rangle \ldots \langle y, A\zeta^{(2L-1)}\rangle\ y\Big]$$

and consider the unit vector along v. Let us consider a "correct" guess of $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$ where all of them are from the random portion, and their supports all contain a fixed co-ordinate (say coordinate 1); note that values of the non-zero entries of $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$ are still random. We show that with at least a constant probability (over the choice of the non-zero entries of $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$), the vector $v = q_1 A_1 + \tilde{v}$ where $\|\tilde{v}\|_2 = o(q_{\max}/\log m)$. Here q_i is the fraction of samples x with i in its support and $q_{\max} = \max_i q_i$. Hence, by running over all possible $\binom{N}{2L-1}$ tuples we can hope to produce candidate vectors that are good approximations to frequently appearing columns of A. Notice that any spurious vectors that we produce will be automatically discarded by our test procedure. While the above algorithm is simple, its analysis requires several new technical ideas including improved concentration bounds for polynomials of *rarely occurring* random variables. To see this, we note that vector v can be written as $v = \sum_{i\in[m]} \gamma_i A_i$ where for each $i \in [m], \gamma_i =$

$$\sum_{J\in[m]^{2L-1}} \sum_{I\in[m]^{2L-1}} \mathop{\mathbb{E}}_{x}\left[x_i x_{i_1} \ldots x_{i_{2L-1}}\right] \prod_{\ell\in[2L-1]} M_{i_\ell, j_\ell}\zeta^{(\ell)}_{j_\ell}.$$

Here M denote the matrix $A^T A$.

We will prove that a constant probability over the randomness in $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$, $\|\sum_{i\neq 1}\gamma_i A_i\|_2 = o(q_{\max}/\log m)$. Notice that since the value distribution is symmetric with mean zero, $\mathbb{E}_x\left[x_i x_{i_1} \ldots x_{i_{2L-1}} x_i\right]$ is zero unless each of the indices in $[m]$ appears an even number of times in $i_1, i_2, \ldots i_{2L-1}$. Hence each γ_i can be further written as the sum of $L^{O(L)}$ many polynomials over $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$, where each polynomial corresponds to a valid partition of the indices $i_1, i_2, \ldots i_{2L-1}$. We will prove that $\|\sum_{i\neq 1}\gamma_i A_i\|_2$ is small by showing concentration bounds for the polynomials corresponding to these terms in γ_i, along with a union bound over the $mL^{O(L)}$ terms. Furthermore these terms are multilinear polynomials of random variables $\zeta^{(1)}, \ldots, \zeta^{(2L-1)}$ that are *rarely occurring* mean-zero random variables i.e., they are non-zero with probability roughly $p = k/m$.

Concentration bounds for multilinear degree-d polynomials of $O(1)$ hypercontractive random variables are known, giving bounds of the form $\mathbb{P}[g(x) > t\|g\|_2] \leq \exp(-ct^{2/d})$ [20]. More recently, sharper bounds (analogous to the Hanson-Wright inequality for quadratic forms [21]) that do not necessarily incur a d factor in the exponent and get bounds of the form $\exp(-\Omega(t^2))$ have also been obtained by Latala, Adamczak and Wolff [22], [23] for sub-gaussian random variables and more generally, random variables of bounded Orlicz ψ_2 norm. However, these seem to give sub-optimal bounds for rarely occurring random variables, as we demonstrate below. On the other hand, bounds that apply in the rarely occurring regime [24], [25] typically apply to polynomials of non-negative random variables with non-negative coefficients, and do not seem directly applicable in our settings.

There are several different terms that arise in these calculations; we give an example of one such term to motivate the need for better concentration bounds in this setting with rarely occurring random variables. One of the terms that arises in the expansion of γ_i is

$$\begin{aligned} Z &= \sum_{j_1, j_2\in[m]} B_{j_1,j_2}\zeta^{(1)}_{j_1}\zeta^{(2)}_{j_2} \\ &:= \sum_{i\in[m]} \sum_{j_1, j_2\in[m]\setminus\{i\}} M_{ij_1}M_{ij_2}\zeta^{(1)}_{j_1}\zeta^{(2)}_{j_2}. \end{aligned}$$

Using the fact that the columns of A are incoherent, for this quadratic form we get that $\|B\|_F = \widetilde{\Omega}(\sqrt{m})$. We can then apply Hanson-Wright inequality to this quadratic form, and conclude that the $|Z| \leq \sqrt{m}\text{poly}\log(n)$ with high probability[9]. On the other hand, the ζ random variables are non-zero with probability at most $p = k/m$ and are $\tau = O(1)$-negatively correlated, and hence we get that $\text{Var}[Z] \leq m\sigma^4(k/m)^2 = \widetilde{O}(k^2/m)$ (and $\mathbb{E}[Z] = 0$). Here σ is the spectral norm of A. Hence, in the ideal case, we can hope to show a much better upper bound of $|Z| \leq k\text{poly}\log(n)/\sqrt{m}$ (smaller by a factor of k/m). Obtaining bounds that take advantage of the small probability of occurrence seems crucial in handling $k = \Omega(\sqrt{m})$ for the semirandom case, and $k = \omega(\sqrt{m})$ for the random case.

To tackle this, we derive general concentration inequalities for multilinear degree-d polynomials of rarely occurring random variables. Let f be a degree d multilinear polynomial

[9]The random variables $\zeta^{(\ell)}_j$ has its ψ_2 Orlicz-norm bounded by $K \leq \log(1/p) = O(\log m)$; Hanson-Wright inequality shows that $\mathbb{P}[|Z| > t] \leq \exp\big(-c\min\big\{\frac{t^2}{K^4\|B\|_F^2}, \frac{t}{K^2\|B\|}\big\}\big)$. Using Hypercontractivity for these distributions also gives similar bounds up to $\text{poly}\log n$ factors.

of the form

$$f(\zeta^{(1)},\dots,\zeta^{(d)}) = \sum_{(j_1,\dots,j_d)\in[m]^d} T_{j_1,\dots,j_d}\zeta^{(1)}_{j_1}\dots\zeta^{(d)}_{j_d},$$

where each of the random variables ζ_j are independent, bounded and non-zero with probability at most p, and for any $\Gamma \subset [d]$. Then we prove that for any $\eta > 0$,

$$\mathbb{P}\left[|f(\zeta^{(1)},\dots,\zeta^{(d)})| \geq \log(2/\eta)^d\sqrt{\rho}\cdot p^{d/2}\|T\|_F\right] \leq \eta, \tag{3}$$

where ρ is a measure of how well-spread out the corresponding tensor T is: it depends in particular, on the maximum row norm ($\|\cdot\|_{2\to\infty}$ operator norm) of different "flattenings" of the tensor T into matrices. This is reminiscent of how the bounds of Latala [22], [23] depends on the spectral norm of different "flattenings" of the tensor into matrices, but they arise for different reasons. We defer to the full version of the paper for a formal statement and more background. To the best of our knowledge, we are not aware of similar concentration bounds for arbitrary multilinear (with potentially non-negative co-efficients) for rarely occurring random variables, and we believe these bounds may be of independent interest in other sparse settings.

The candidate generation algorithm works for sparsity of $k = \widetilde{O}(\sqrt{n})$ in the semirandom case, and a better sparsity of $k = \widetilde{O}(n^{2/3})$ in the random case. The analysis for both the semirandom case and random case proceeds by carefully analyzing various terms that arise in evaluating $\{\gamma_i : i \in [m]\}$, and using the new concentration bounds in the context of each of these terms along with good bounds on the norms of various tensors and their flattenings that arise (this uses sparsity of the samples, the incoherence assumption and the spectral norm bound among other things). We now describe one of the simpler terms that arise in the random case, to demonstrate the advantage of considering larger L i.e., more fixed samples. Consider the expression

$$Z = \sum_{J\in[m]^{2L-1}} M_{i,j_{2L-1}}\zeta^{(2L-1)}_{j_{2L-1}} \sum_{i_1,\dots,i_{L-1}} \mathbb{E}\left[x_i^2 x_{i_1}^2\dots x_{i_{L-1}}^2\right] \times \prod_{\ell\in[L-1]} M_{i_\ell,j_{2\ell-1}}M_{i_\ell,j_{2\ell}}\zeta^{(2\ell-1)}_{j_{2\ell-1}}\zeta^{(2\ell)}_{j_{2\ell}}. \tag{4}$$

In the random case, $\mathbb{E}[x_i^2 x_{i_1}^2\dots x_{i_{L-1}}^2] \approx \mathbb{E}[x_i^2]\,\mathbb{E}[x_{i_1}^2]\dots\mathbb{E}[x_{i_{L-1}}^2] \leq (k/m)^L$, since the support distribution is essentially random (this also assumes the value distribution is Rademacher). Further, for the corresponding tensor T of co-efficients, one can show a bound of $\|T\|_F = O\big(m^{(L-1)/2}\big)$. Hence, applying the new concentration bound we would get an ideal bound (assuming the imbalance factor $\rho = O(1)$) of roughly $c\cdot(k/m)^L\sqrt{m}^{L-1}\cdot(k/m)^{L-1/2} = c\big(\frac{k^2}{m\sqrt{m}}\big)^{L-1}\cdot(k/m)^{3/2}$, which becomes $o(k/(m\sqrt{m}))$ as required for L being a sufficiently large constant when $k = o(m^{3/4-\varepsilon})$ [10]. On the other hand, with higher values of L there are some lower-order terms that start becoming larger comparatively, for which the new concentration bounds for polynomials of rarely occurring random variables becomes critical. Balancing out these terms allows us to handle a sparsity of $k = \widetilde{O}(n^{2/3})$ for the random case.

The semirandom model presents several additional difficulties as compared to the random model. Firstly, as most of the data is generated with arbitrary supports, we cannot assume that the x variables are $\tau = O(1)$-negatively correlated. As a result, the term $\mathbb{E}[x_i^2 x_{i_1}^2\dots x_{i_{L-1}}^2]$ does not factorize as the adversary can make the joint probability distribution of the non-zeros very correlated. Hence, to bound various expressions that appear in the expansion of γ_i, we need to use inductive arguments to upper bound the magnitude of each inner sum and eliminating the corresponding running index (this needs to be done carefully since these quantities can be negative). We bound each inner sum using the new concentration bounds for polynomials of rarely occurring random variables repeatedly along with the inequality $\sum_{i_d\in[m]}\mathbb{E}[x_i^2 x_{i_1}^2\dots x_{i_d}^2] \leq k\,\mathbb{E}[x_i^2 x_{i_1}^2\dots x_{i_{d-1}}^2]$, and some elegant linear algebraic facts.

Finally, the above procedure can be used to recover all the columns A_i of the dictionary whose corresponding occurrence probabilities $q_i = \mathbb{E}[x_i^2]$ are close to the largest i.e., $q_i = \widetilde{\Omega}(\max_{j\in[m]} q_j)$. To recover all the other columns, we use a linear program and subsample the data (just based on columns recovered so far), so that one of the undiscovered columns has largest occurrence probability. We defer to the details in the full version of the paper.

C. Related Work

Polynomial Time Algorithms: Spielman et al. [8] were the first to provide a polynomial time algorithm with rigorous guarantees for dictionary learning. They handled the full rank case, i.e, $m = n$, and assumed the following distributional assumptions about X: each entry is chosen to be non-zero independently with probability $k/m = O(1)/\sqrt{n}$ (the support distribution $\mathcal{D}^{(s)}$ is essentially uniformly random) and conditioned on the support, each non-zero value is set independently at random from a sub-Gaussian distribution e.g., Rademacher distribution (the value distribution $\mathcal{D}^{(v)}$). Their algorithm uses the insight that w.h.p. in this model, the sparsest vectors in the row space of Y correspond to the rows of X, and solve a sequence of LPs to recover X and A. Subsequent works [26], [27], [9] have focused on improving the sample complexity and sparsity assumptions in the full-rank setting. However in the presence of the semirandom adversary, the sparsest vectors in the row space of Y may

[10]The bound that we actually get in this case is off by a $c = \sqrt{m}\mathrm{poly}\log n$ factor since $\rho = \omega(1)$, but this also becomes small for large L.

not contain rows of X and hence the algorithmic technique of [8] breaks down.

For the case of over-complete dictionaries the works of Arora et al. [10] and Agarwal et al. [11] provided polynomial time algorithms when the dictionary A is μ-incoherent. In particular, the result of [10] also holds under a weaker assumption that the support distribution $\mathcal{D}^{(s)}$ is approximately $\ell = O(1)$-wise independent i.e., $\mathbb{P}_{x\sim\mathcal{D}^{(s)}}[i_1, i_2, \ldots, i_\ell \in \text{supp}(x)] \leq \tau^\ell (k/m)^\ell$ for some constant $\tau > 0$. Under this assumption they can handle sparsity up to $\widetilde{O}(\min(\sqrt{n}, m^{1/2-\varepsilon}))$ for any constant $\varepsilon > 0$ with $\ell = O(1/\varepsilon)$. Their algorithm computes a graph G over the samples in Y by connecting any two samples that have a high dot product – these correspond to pairs of samples whose supports have at least one column in common. Recovering columns of A then boils down to identifying communities in this graph with each community identifying a column of A. Subsequent works have focused on extending this approach to handle mildly weaker or incomparable assumptions on the dictionary A or the distribution of X [28], [12]. For example, the algorithm of [12] only assumes $O(1)$-wise independence on the non-zero values of a column x. The state of the art results along these lines can handle $k = \widetilde{O}(\sqrt{n})$ sparsity for $\mu = \widetilde{O}(1)$-incoherent dictionaries. Again, we observe that in the presence of the semirandom adversary, the community structure present in the graph G could become very noisy and one might not be able to extract good approximations to the columns of A, or worse still, find spurious columns.

The work of Barak at al. [13] reduce the problem of recovering the columns of A to a (noisy) tensor decomposition problem, which they solve using Sum-of-Squares (SoS) relaxations. Under assumptions that are similar to that of [10] (assuming approximate $\widetilde{O}(1)$-wise independence), these algorithms based on SoS relaxations [13], [14] handle almost linear sparsity $k = \widetilde{O}(n)$ and recover incoherent dictionaries with quasi-polynomial time guarantees in general, and polynomial time guarantees when $\sigma = O(1)$ (this is obtained by combining Theorem 1.5 in [14] with [13]). The recent work of Kothari et al. [29] also extended these algorithms based on tensor decompositions using SoS, to a setting when a small fraction of the data can be adversarially corrupted or arbitrary. This is comparable to the setting in the semirandom model when $\beta = 1 - \varepsilon$ (for a sufficiently small constant ε), but the non-zero values for these samples can also be arbitrary. However in the semirandom model, the reduction from dictionary learning to tensor decompositions breaks down because the supports can have arbitrary correlations in aggregate, particularly when β is small. Hence these algorithms do not work in the semirandom model.

Moreover, even in the absence of any adversarial samples, Theorem I.2 and the current state-of-the-art guarantees [14], [12] are incomparable, and are each optimal in their own setting. For instance, consider the setting when the over-completeness $m/n, \sigma = O(n^\varepsilon)$ for some small constant $\varepsilon > 0$. In this case, Arora et al. [12] can handle a sparsity of $\widetilde{O}(\sqrt{n})$ in polynomial time and Ma et al. [14] handle $\widetilde{O}(n)$ sparsity in quasi-polynomial time, while Theorem I.2 handles a sparsity of $\widetilde{O}(n^{2/3})$ in polynomial time. On the other hand, [12] has a better dependence on σ, while [14] can handle $\widetilde{O}(n)$ sparsity when $\sigma = O(1)$. Further, both of these prior works do not need full independence of the value distribution $\mathcal{D}^{(v)}$ and the SoS-based approaches work even under mild incoherence assumptions to give some weak recovery guarantees[11] However, we recall that in addition our algorithm works in the semirandom model (almost arbitrary support patterns) up to sparsity $\widetilde{O}(\sqrt{n})$, and this seems challenging for existing algorithms.

Heuristics and Associated Guarantees: Many iterative heuristics like k-SVD, method of optimal direction (MOD), and alternate minimization have been designed for dictionary learning, and recently there has also been interest in giving provable guarantees for these heuristics. Arora et al. [10] and Agarwal et al. [30] gave provable guarantees for k-SVD and alternate minimization assuming initialization with a close enough dictionary. Arora et al. [12] provided guarantees for a heuristic that at each step computes the current guess of X by solving sparse recovery, and then takes a gradient step of the objective $\|Y - AX\|^2$ to update the current guess of A. They initialize the algorithm using a procedure that finds the principal component of the matrix $E[\langle u^{(1)}, y\rangle \langle u^{(2)}, y\rangle\ yy^T]$ for appropriately chosen samples $u^{(1)}, u^{(2)}$ from the data set. A crucial component of our algorithm in the semirandom model is a procedure to generate candidate vectors for the columns of A and is inspired by the initialization procedure of [12].

Identifiability Results: As with many statistical models, most identifiability results for dictionary learning follow from efficient algorithms. As a result identifiability results that follow from the results discussed above rely on strong distributional assumptions. On the other hand results establishing identifiability under deterministic conditions [31], [32] require exponential sample complexity as they require that every possible support pattern be seen at least once in the sample, and hence require $O(m^k)$ samples. To the best of our knowledge, our results (Theorem I.4) lead to the first identifiability results with polynomial sample complexity without strong distributional assumptions on the supports.

Other Related Work: A problem which has a similar flavor to dictionary learning is Independent Component Analysis (ICA), which has been a rich history in signal processing and computer science [33], [34], [35]. Here, we are given $Y = AX$ where each entry of the matrix X is independent, and there are polynomial time algorithms both in the under-complete [34] and over-complete case [36], [35]

[11]However, to recover A and X to high accuracy, incoherence and RIP assumptions of the kind assumed in our work and [12] seem necessary.

that recover A provided each entry of X is non-Gaussian. However, these algorithms do not apply in our setting, since the entries in each column of X are not independent (the supports can be almost arbitrarily correlated because of the adversarial samples).

Finally, starting with the works of Blum and Spencer [15], semirandom models have been widely studied for various optimization and learning problems. Feige and Kilian [16] considered semi-random models involving monotone adversaries for various problems including graph partitioning, independent set and clique. Semirandom models have also been studied in the context of unique games [37], graph partitioning problems [38], [39] and learning communities [40], [41], [42], correlation clustering [43], [44], noisy sorting [45], coloring [46] and clustering [47].

II. Preliminaries

We will use A to denote an $n \times m$ over-complete ($m > n$) dictionary with columns $A_1, A_2, \ldots A_m$. Given a matrix or a higher order tensor T, we will uses $\|T\|_F$ to denote the Frobenius norm of the tensor. For matrices A we will use $\|A\|_2$ to denote the spectral norm of A. We first define the standard random model for generating data from an over-complete dictionary.

Informally, a vector $y = Ax$ is generated as a random linear combination of a few columns of A. We first pick the support of x according to a *support distribution* denoted by $\mathcal{D}^{(s)}$, and then draw the values of each of the non-zero entries in x independently according to the *value distribution* denoted by $\mathcal{D}^{(v)}$. $\mathcal{D}^{(s)}$ is a distribution that is over the set of vectors in $\{0,1\}^m$ with at most k ones.

Value Distribution:: As is standard in past works on sparse coding [10], [12], we will assume that the value distribution $\mathcal{D}^{(v)}$ is any mean zero symmetric distribution supported in $[-C,-1] \cup [1,C]$ for a constant $C > 1$. This is known as the *Spike-and-Slab* model [18]. For technical reasons we also assume that $\mathcal{D}^{(v)}$ has non-negligible density in $[1, 1+\eta]$ for $\eta = 1/(\operatorname{poly}\log n)$. Formally we assume that

$$\exists \gamma_0 \in (0,1) \text{ s.t. } \forall \eta \geq \frac{1}{\log^c n}, \mathbb{P}_{\mathcal{D}^{(v)}}([1,1+\eta]) \geq \gamma_0. \quad (5)$$

In the above definition, we will think of γ_0 as just being non-negligible (e.g., $1/\operatorname{poly}(n)$). This assumption is only used in Section III, and the sample complexity will only involve inverse polynomial dependence on γ_0. The above condition captures the fact that the value distribution has some non-negligible mass close to 1 [12]. Further, this is a benign assumption that is satisfied by many distributions including the Rademacher distribution that is supported on $\{+1,-1\}$ (with $\gamma_0 = 1/2$), and the uniform distribution over $[-C,-1] \cup [1,C]$ (with $\gamma_0 = 1/(2C)$).

Random Support Distribution $\mathcal{D}_R^{(s)}$.: Let $\xi \in \mathbb{R}^m$ be drawn from $\mathcal{D}_R^{(s)}$. To ensure that each column appears reasonably often in the data so that recovery is possible information theoretically we assume that each coordinate i in ξ is non-zero with probability $\frac{k}{m}$. We do not require the non-zero coordinates to be picked independently and there could be correlations provided that they are negatively correlated up to a slack factor of τ.

Definition II.1. For any $\tau \geq 1$, a set of non-negative random variables $Z_1, Z_2, \ldots, Z_m$ where $P(Z_i \neq 0) \leq p$ is called τ-negatively correlated if for any $i \in [m]$ and any $S \subseteq [m]$ such that $i \notin S$ and $|S| = O(\log m)$ we have that for a constant $\tau > 0$,

$$P\big(Z_i \neq 0 \big| \bigcap_{j \in S} Z_j \neq 0\big) \leq \tau p. \quad (6)$$

In the random model the variables $\xi_1, \xi_2, \ldots, \xi_m$ are τ-negatively correlated with $p = \frac{k}{m}$. We remark that for our algorithms we only require the above condition (for the random portion of the data) to hold for sets S of size up to $O(\log m)$. Of course in the semi-random model described later, the adversary can add additional data from supports distributions with arbitrary correlations; hence they are not τ-negatively correlated, and each co-ordinate of x need not be non-zero with probability at most $p = k/m$.

Random model for Dictionary Learning.: Let $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$ denote the distribution over $\mathbb{R}^m$ obtained by first picking a support vector from $\mathcal{D}_R^{(s)}$ and then independently picking a value for each non zero coordinate from $\mathcal{D}^{(v)}$. Then we have that a sample y from the over complete dictionary is generated as

$$y = \sum_{i \in [m]} x_i A_i,$$

where $(x_1, x_2, \ldots, x_m)$ is generated from $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$. Given $S = \{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\}$ drawn from the model above, the goal in standard dictionary learning is to recover the unknown dictionary A^*, up to signs and permutations of columns.

A. Semi-random model

We next describe the *semi-random* extension of the above model for sparse coding. In the semi-random model an initial set of samplesis generated from the standard model described above. A semi-random adversary can then an arbitrarily number of additional samples with each sample $y = Ax$ generated by first picking the support of x arbitrarily and then independently picking values of the non-zeros according to $\mathcal{D}^{(v)}$. Formally we have the following definition

Definition II.2 (Semi-Random Model: $\mathcal{M}_\beta(\mathcal{D}_R^{(s)}, \widetilde{\mathcal{D}}^{(s)}, \mathcal{D}^{(v)})$)**.** A semi-random model for

[12] If the value distribution has negligible mass in $[1,1+\eta] \cup [-1-\eta,-1]$, one can arguably rescale the value distribution by $(1+\eta)$ so that all of the value distribution is essentially supported on $[1, C/(1+\eta)] \cup [-C/(1+\eta), -1]$.

sparse coding, denoted as $\mathcal{M}_\beta(\mathcal{D}_R^{(s)}, \widetilde{\mathcal{D}}^{(s)}, \mathcal{D}^{(v)})$, is defined via the following process of producing N samples

1) Given a τ-negatively correlated support distribution $\mathcal{D}_R^{(s)}$, $N_0 = \beta N$ "random" support vectors $\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(N_0)}$ are generated from $\mathcal{D}_R^{(s)}$.
2) Given the knowledge of the supports of $\xi^{(1)}, \ldots, \xi^{(N_0)}$, the semi-random adversary generates $(1-\beta)N$ additional support vectors $\xi^{(N_0+1)}, \xi^{(N_0+2)}, \ldots, \xi^{(N)}$ from an arbitrary distribution $\widetilde{\mathcal{D}}^{(s)}$. The choice of $\widetilde{\mathcal{D}}^{(s)}$ can depend on $\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(N_0)}$.
3) Given a value distribution $\mathcal{D}^{(v)}$ that satisfies the Spike-and-Slab model, the vectors $x^{(1)}, x^{(2)}, \ldots, x^{(N_0)}, x^{(N_0+1)}, \ldots, x^{(N)}$ are form by picking each non-zero value (as specified by $\xi^{(1)}, \ldots, \xi^{(N)}$ respectively) independently from the distribution $\mathcal{D}^{(v)}$.
4) $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$ are randomly reordered as columns of an $m \times N$ matrix X. Then the output of the model is $Y = AX$.

We would like to stress that the amount of semi-random data can overwhelm the initial random set. In other words, β need not be a constant and can be a small inverse polynomial factor. The number of samples needed for our algorithmic results will have an inverse polynomial dependence on β. While the above description of the model describes a distribution from which samples can be drawn, one can also consider a setting where there a fixed number of samples N, of which $\beta N = N_0$ samples were drawn with random supports i.e., from $\mathcal{D}_R^{(s)}$. These two descriptions are essentially equivalent in our context since the distribution $\widetilde{\mathcal{D}}^{(s)}$ is arbitrary. However, since there are multiple steps in the algorithm, it will be convenient to think of this as a generative distribution that we can draw samples from (in the alternate view, we can randomly partition the samples initially with one portion for each step of the algorithm).

In the next few sections we give the formal statements of our main results. We defer to the full version of the paper [48] for proofs and all the details.

III. Testing Procedure and Identifiability

In this section we describe and prove the correctness of our testing procedure that checks if a given unit vector z is close to any column of the dictionary A. The procedure works as follows: it takes a value η as input and checks if the inner product $|\langle z, Ax\rangle|$ only takes values in $[0,\eta] \cup [1-\eta, C(1+\eta)]$ for most samples x, and if $|\langle z, Ax\rangle| \in [1-\eta, C(1+\eta)]$ for a non-negligible fraction of samples. In other words, a vector z is rejected only if $|\langle z, Ax\rangle| \in (2\eta, 1-2\eta)$ for a non-negligible fraction of the samples, or if $|\langle z, Ax\rangle| \in [1-\eta, C(1+\eta)]$ for a negligible fraction of samples. For any $\eta \in (0,1)$, we will often use the notation I_η to denote the set $\{\, t \in \mathbb{R} : |t| \in [1-\eta, C(1+\eta)] \cup [0,\eta] \,\}$, i.e. the range of values close to 0 or 1.

Algorithm **TestColumn**$(z, Y = \{\, y^{(1)}, \ldots, y^{(N)} \,\}, \kappa_0, \kappa_1, \eta)$

1) Let $\widetilde{\kappa}_1$ be the fraction of samples such that $|\langle z, y^{(r)}\rangle| \in [1-\eta, C(1+\eta)]$ and $\widetilde{\kappa}_0$ be the fraction of samples such that $|\langle z, y^{(r)}\rangle| \notin [1-\eta, C(1+\eta)] \cup [0, C\eta]$.
2) If $\widetilde{\kappa}_0 < \kappa_0$ and $\widetilde{\kappa}_1 \geq \kappa_1$, return $(\text{YES}, \widehat{z})$, where $z' = \text{mean}\big(\{\, y^{(r)} : r \in [N] \text{ s.t. } \langle y^{(r)}, z\rangle \geq \frac{1}{2} \,\}\big)$ and $\widehat{z} = z'/\|z'\|_2$.
3) Else return $(\text{NO}, \emptyset)$.

Figure 1.

We show the following guarantees for Algorithm TestColumn. We will prove the guarantees in a slightly broader setup so that it can be used both for the identifiability results and for the algorithmic results. We assume that we are given N samples $\{\, y^{(r)} = Ax^{(r)} : r \in [N] \,\}$, when the value distribution (distribution of each non-zero co-ordinate of a given sample $x^{(r)}$) is given by $\mathcal{D}^{(v)}$.We make the following mild assumption about the sparsity pattern (support); for any i and any $T \subset [m] \setminus \{\, i \,\}$, we assume that there are at least $q_{\min} N$ samples which contain i but do not contain T in the support. Note that for the semi-random model, if β fraction of the samples come from the random portion, then $q_{\min} \geq \frac{1}{2}\beta k/m$ with high probability.

In what follows, it will be useful to think of $\eta = O(1/\text{poly}\log(n))$, $\gamma_0 = n^{-\Omega(1)}$, the desired accuracy $\eta_0 = 1/\text{poly}(n)$, sparsity $k = O(n/\text{poly}\log(n))$, and the desired failure probability to be $\gamma = \exp(-n)$. Hence, in this setting $\kappa_0 = n^{-\Omega(1)}$ and $\delta = O(1/\text{poly}\log(n))$ as well.

Theorem III.1 (Guarantees for TestColumn)**.** *There exists constants $c_0, c_1, c_2, c_3, c_4, c_5 > 0$ (potentially depending on C) such that the following holds for any $\gamma \in (0,1), \eta_0 < \eta \in (0,1)$ satisfying $\sqrt{\frac{c_3 k}{m}} < \eta < \frac{c_1}{\log^2\left(\frac{mn}{q_{\min}\eta_0}\right)}$. Set $\kappa_0 := c_4\gamma_0\eta q_{\min}/(km)$. Suppose we are given $N \geq \frac{c_2 knm\log(1/\gamma)}{\eta_0^3\gamma_0\kappa_0}$ samples $y^{(1)}, \ldots, y^{(N)}$ satisfying*

- *the dictionary A is (k,δ)-RIP for $\delta < \left(\frac{\eta}{16C\log(1/\kappa_0)}\right)^2$,*
- *$\forall i \in [m], T \subset [m] \setminus \{\, i \,\}$ with $|T| \leq c_3/\eta^2$, there at least $q_{\min} N$ samples whose supports all contain i, but disjoint from T.*

Suppose we are given a unit vector $z \in \mathbb{R}^n$, then TestColumn$(z, \{\, y^{(1)}, \ldots, y^{(N)} \,\}, 2\kappa_0, \kappa_1 = c_5 q_{\min}\gamma_0\eta, \eta)$ *runs in times $O(N)$ time, and we have with probability at least $1-\gamma$ that*

- (Completeness) *if $\|z - bA_i\|_2 \leq \eta' = \eta/(8C\log(1/\kappa_0))$ for some $i \in [m], b \in \{\,-1, 1\,\}$, then Algorithm* TestVector *outputs $(\text{YES}, \widehat{z})$.*
- (Soundness) *if unit vector $z \in \mathbb{R}^n$ passes* TestColumn*, then there exists $i \in [m], b \in \{\,-1, 1\,\}$ such that $\|z - bA_i\|_2 \leq \sqrt{8\eta}$. Further, in this case $\|\widehat{z} - bA_i\|_2 \leq \eta_0$.*

Remark III.2. We note that the above algorithm is also robust to adversarial noise. In particular, if we are given samples of the form $y^{(r)} = Ax^{(r)} + \psi^{(r)}$, where $\|\psi^{(r)}\|_2 \leq O(\eta)$, then it is easy to see that the completeness and soundness guarantees go through since the contribution to $\langle y^{(r)}, z\rangle$ is at most $|\langle \psi^{(r)}, z\rangle| \leq \|\psi^{(r)}\| = O(\eta)$.

The above theorem immediately implies an identifiability result for the same model (and hence the semi-random model). By applying Algorithm TESTCOLUMN to each z in an $\widetilde{\Omega}(\eta)$-net over $\mathbb{R}^n$ dimensional unit vectors and choosing $\gamma = \exp\big(-\Omega(n\log(1/\eta))\big)$ in Theorem III.1 and performing a union bound over every candidate vector z in the net, we get the following identifiability result as long as $k < n/\mathrm{poly}\log(n)$.

Corollary III.3 (Identifiability for Semi-random Model)**.** *There exists constants $c_0, c_1, c_2, c_3, c_4, c_5, c_6 > 0$ (potentially depending on C) such that the following holds for any $k < n/\log^{2c_1} m$, $\eta_0 \in (0,1)$. Set $\kappa_0 := c_0\gamma_0 \log^{-c_1} m q_{\min}$. Suppose we are given $N \geq \frac{c_2 knm \log^{c_1} m \log(1/\kappa_0)}{\eta_0^3 \gamma_0 q_{\min}}$ samples $y^{(1)}, \ldots, y^{(N)}$ satisfying*

- *the dictionary A is (k,δ)-RIP for $\delta < \frac{c_5}{\log(1/\kappa_0)\log^{c_6} m}$,*
- *$\forall i \in [m], T \subset [m] \setminus \{i\}$ with $|T| \leq c_4 \log^{2c_1} m$, there at least $q_{\min} N$ samples whose supports all contain i, but disjoint from T.*

Then there is an algorithm that with probability at least $1-\exp(-n)$ finds the columns $\widehat{A}$ such that $\|\widehat{A}_i - b_i A_i\|_2 \leq \eta_0$ for some $b \in \{-1,1\}^m$.

IV. STRONGER IDENTIFIABILITY FOR RADEMACHER VALUE DISTRIBUTION

In the special case when the value distribution is a Rademacher distribution (each x_i is $+1$ or -1 with probability $1/2$ each), we can obtain even stronger guarantees for the testing procedure. We do not need to assume that there are non-negligible fraction of samples $y = Ax$ where the support distribution is "random" [13]. Here, we just need that for every triple $i_1, i_2, i_3 \in [m]$ of columns, they jointly occur in at least a non-negligible number of samples.

On the other hand, we remark that the triple co-occurrence condition is arguably the weakest condition under which identifiability is possible. In the full version we show a non-identifiability statement even when the value distribution is a Rademacher distribution. In this example, for every pair of columns there are many samples where these two columns co-occur.

Theorem IV.1 (Rademacher Value Distribution)**.** *There exists constants $c_0, c_1, c_2, c_3, c_4 > 0$ such that the following holds for any $\gamma \in (0,1), \eta_0 < \eta \in (0,1)$ satisfying $\sqrt{\frac{c_3 k}{m}} < \eta < \frac{c_1}{\log^2\left(\frac{mn}{q_0\eta_0}\right)}$. Set $\kappa_0 := c_4\eta q_0/(km)$. Suppose we are given $N \geq \frac{c_2 knm\log(1/\gamma)}{\eta_0^3\kappa_0}$ samples $y^{(1)}, \ldots, y^{(N)}$ satisfying*

- *the dictionary A is (k,δ)-RIP for $\delta < \left(\frac{\eta}{16\log(1/\kappa_0)}\right)^2$,*
- *$\forall i_1, i_2, i_3 \in [m]$, there at least $q_0 N$ samples whose supports all contain i_1, i_2, i_3.*

Then there is an algorithm TESTCOL_RAD *such that given a unit vector $z \in \mathbb{R}^n$,* TESTCOL_RAD *called with parameters $(z, \{y^{(1)}, \ldots, y^{(N)}\}, 2\kappa_0, \kappa_1 = c_5\eta q_0, \eta)$ runs in times $O(N)$ time, and we have with probability at least $1-\gamma$ that*

- (Completeness) *if $\|z - bA_i\|_2 \leq \eta' = \eta/(8\log(1/\kappa_0))$ for some $i \in [m], b \in \{-1,1\}$, then the algorithm outputs (YES, z').*
- (Soundness) *if unit vector $z \in \mathbb{R}^n$ passes the algorithm then there exists $i \in [m], b \in \{-1,1\}$ such that $\|z - bA_i\|_2 \leq \sqrt{8\eta}$. Further, in this case $\|z' - bA_i\|_2 \leq \eta_0$.*

As before, we note that the above algorithm is also robust to adversarial noise of the order of magnitude $O(\eta)$ in every sample. Further, the above theorem again implies an identifiability result by applying it to each candidate unit vector z in an $\widetilde{\Omega}(\eta)$-net over $\mathbb{R}^n$ dimensional unit vectors and choosing $\gamma = \exp\big(-\Omega(n\log(1/\eta))\big)$ for $k < n/\mathrm{poly}\log(n)$.

Corollary IV.2 (Identifiability for Rademacher Value Distribution)**.** *There exists constants $c_0, c_1, c_2, c_3, c_4, c_5, c_6 > 0$ such that the following holds for any $k < n/\log^{2c_1} m$, $\eta_0 \in (0,1)$. Set $\kappa_0 := c_0 \log^{-c_1} m q_0$. Suppose we are given $N \geq c_2 knm\eta_0^{-3} q_0^{-1} \log^{c_1} m \log(1/\kappa_0)$ samples $y^{(1)}, \ldots, y^{(N)}$ satisfying*

- *the dictionary A is (k,δ)-RIP for $\delta < \frac{c_5}{\log(1/\kappa_0)\log^{c_6} m}$,*
- *$\forall i_1, i_2, i_3 \in [m]$, there at least $q_0 N$ samples whose supports all contain i_1, i_2, i_3.*

Then there is an algorithm that with probability at least $1-\exp(-n)$ finds the columns $\widehat{A}$ (up to renaming columns) such that $\|\widehat{A}_i - b_i A_i\|_2 \leq \eta_0$ for some $b \in \{-1,1\}^m$.

The test procedure for checking whether unit vector z is close to a column is slightly different. In addition to Algorithm TESTCOLUMN, there is an additional procedure that is less stringent: it checks if the inner product $|\langle z, Ax\rangle|$ only takes values in $[0,\eta] \cup [1-\eta, C(1+\eta)]$ for most samples $x \sim \mathcal{D}$, and if $|\langle z, Ax\rangle| \in [1-\eta, C(1+\eta)]$ for a non-negligible fraction of samples. In other words, a vector z is rejected only if $|\langle z, Ax\rangle| \in (2\eta, 1-2\eta)$ for a non-negligible fraction of the samples, or if $|\langle z, Ax\rangle| \in [1-\eta, C(1+\eta)]$ for a negligible fraction of samples.

V. EFFICIENT ALGORITHMS FOR PRODUCING CANDIDATE COLUMNS

The main theorem of this section is a polynomial time algorithm for recovering incoherent dictionaries when the samples come from the semirandom model.

[13] In particular, we don't need to assume for any $i, T \subseteq [m] \setminus \{i\}$ of small size, that we have many samples that contain i but not T.

Theorem V.1. *Let A be a μ-incoherent $n \times m$ dictionary with spectral norm σ. There is an algorithm* RECOVERDICT *such that for any $\varepsilon > 0$, given $N = \mathrm{poly}(k, m, n, 1/\varepsilon, 1/\beta)$ samples from the semi-random model $\mathcal{M}_\beta(\mathcal{D}_R^{(s)}, \widetilde{\mathcal{D}}^{(s)}, \mathcal{D}^{(v)})$, Algorithm* RECOVERDICT *with probability at least $1 - \frac{1}{m}$, outputs a set W^* such that*

- *For each column A_i of A, there exists $\hat{A}_i \in W^*$, $b \in \{\pm 1\}$ such that $\|A_i - b\hat{A}_i\| \le \varepsilon$.*
- *For each $\hat{A}_i \in W^*$, there exists a column A_i of A, $b \in \{\pm 1\}$ such that $\|\hat{A}_i - bA_i\| \le \varepsilon$,*

provided $k \le \sqrt{n}/\nu_1(\frac{1}{m}, 16)$. Here $\nu_1(\eta, d) := c_1 \tau \mu^2 \big(C(\sigma^2 + \mu\sqrt{\frac{m}{n}}) \log^2(n/\eta)\big)^d$, $c_1 > 0$ is a constant (potentially depending on C), and the polynomial bound for N also hides a dependence on C.

The bound above is the strongest when $m = \widetilde{O}(n)$ and $\sigma = \widetilde{O}(1)$, in which case we get guarantees for $k = \widetilde{O}(\sqrt{n})$, where $\widetilde{O}$ also hides dependencies on τ, μ. However, notice that we can also handle $m = O(n^{1+\varepsilon_0}), \sigma = O(n^{\varepsilon_0})$, for a sufficiently small constant ε_0 at the expense of smaller sparsity requirement – in this case we handle $k = \widetilde{O}(n^{1/2 - O(\varepsilon_0)})$ (we do not optimize the polynomial dependence on σ in the above guarantees). The above theorem gives a polynomial time algorithm that recovers the dictionary (up to any inverse polynomial accuracy) as long as β, the fraction of random samples is inverse polynomial. In particular, the sparsity assumptions and the recovery error do not depend on β. In other words, the algorithm succeeds as long we are given a few "random" samples (say N_0 of them), even where there is a potentially a much larger polynomial number $N \gg N_0$ of samples with arbitrary supports. We remark that the above algorithm is also robust to inverse polynomial error in each sample; however we omit the details for sake of exposition.

VI. Efficient algorithms for the random model: Beyond $\sqrt{n}$ sparsity

In this section we show that when the data is generated from the standard random model $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$ our approach from the previous section leads to an algorithm that can handle sparsity up to $\widetilde{O}(n^{2/3})$ which improves upon the state-of-art results in certain regimes, as described in the full version. As in the semi-random case, we will look at the statistic $\mathbb{E}[\langle u^{(1)}, y\rangle \langle u^{(2)}, y\rangle \langle u^{(3)}, y\rangle \dots \langle u^{(2L-1)}, y\rangle\ y]$ for a constant $L \ge 8$. Here $u^{(1)}, u^{(2)}, \dots, u^{(2L-1)}$ are samples that all have a particular column, say A_i, in their support such that A_i appears with the same sign in each sample. Unlike in the semi-random case where one was only able to recover high frequency columns, here we will show that then one can good approximation to any columns A_i via this approach. Hence, in this case we do not need to iteratively re-weigh the data to recover more columns. This is due to the fact that in the random case, given a sample $y = Ax$, we have that $P(x_i \neq 0) = \frac{k}{m}$. Hence, all columns are large frequency columns. Furthermore, when analyzing various sums of polynomials over the ζ random variables we will be able to use better concentration bounds. The main theorem of this section stated below claims that there is an algorithm RECOVERCOLUMNS that will output good approximations to all columns of A when fed with data from the random model $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$.

Theorem VI.1. *There exists constants $c_1 > 0$ (potentially depending on C) and $c_2 > 0$ such that the following holds for any $\varepsilon > 0$, any constants $c > 0$, $L \ge 8$. Let $A_{n \times m}$ be a μ-incoherent matrix with spectral norm at most σ that satisfies (k, δ)-RIP for $\delta < 1/(C^2 \log^{c_2} n)$. Given $\mathrm{poly}(k, m, n, 1/\varepsilon)$ samples from the random model $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$, Algorithm* RECOVERCOLUMNS*, with probability at least $1 - \frac{1}{m^c}$, outputs a set W such that*

- *For each $i \in [m]$, W contains a vector $\hat{A}_i$, and there exists $b \in \{\pm 1\}$ such that $\|A_i - b\hat{A}_i\| \le \varepsilon$.*
- *For each vector $\hat{z} \in W$, there exists A_i and $b \in \{\pm 1\}$ such that $\|\hat{z} - bA_i\| \le \varepsilon$,*

provided $k \le n^{2/3}/(\nu(\frac{1}{m}, 2L)\tau\mu^2)$. Here $\nu(\eta, d) := c_1 \big(C(\sigma^2 + \mu\sqrt{\frac{m}{n}}) \log^2(n/\eta)\big)^d$, and the polynomial bound also hides a dependence on C and L.

Here, we use $\mathcal{D}_R^{(s)} \odot \mathcal{D}^{(v)}$ as the first argument to the RECOVERCOLUMNS procedure and it should be viewed as a model $\mathcal{M}_\beta(\mathcal{D}_R^{(s)}, \mathcal{D}_R^{(s)}, \mathcal{D}^{(v)})$ with $\beta = 1$. Again the bound above is strongest when $m = O(n), \sigma = O(1)$ in which case we get $k \le \widetilde{O}(n^{2/3})$, However, as in the semirandom case, we can handle $m = n^{1+\varepsilon_0}$ for a sufficiently small constant $\varepsilon_0 > 0$ with a weaker dependence on the sparsity.

VII. Acknowledgements

The authors thank Sivaraman Balakrishnan, Aditya Bhaskara, Anindya De, Konstantin Makarychev and David Steurer for several helpful discussions. Aravindan Vijayaraghavan is supported by the National Science Foundation (NSF) under Grant No. CCF-1652491 and CCF-1637585.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[2] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0042698997001697

[3] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[4] D. L. Donoho and P. B. Stark, "Uncertainty principles and signal recovery," *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.

[5] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE transactions on information theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[6] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theor.*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1109/TIT.2005.858979

[7] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[8] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 3087–3090. [Online]. Available: http://dl.acm.org/citation.cfm?id=2540128.2540583

[9] Q. Qu, J. Sun, and J. Wright, "Finding a sparse vector in a subspace: Linear sparsity using alternating directions," in *Advances in Neural Information Processing Systems*, 2014, pp. 3401–3409.

[10] S. Arora, R. Ge, and A. Moitra, "New algorithms for learning incoherent and overcomplete dictionaries," in *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, 2014, pp. 779–806. [Online]. Available: http://jmlr.org/proceedings/papers/v35/arora14.html

[11] A. Agarwal, A. Anandkumar, and P. Netrapalli, "Exact recovery of sparsely used overcomplete dictionaries," *stat*, vol. 1050, pp. 8–39, 2013.

[12] S. Arora, R. Ge, T. Ma, and A. Moitra, "Simple, efficient, and neural algorithms for sparse coding," in *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, 2015, pp. 113–149. [Online]. Available: http://jmlr.org/proceedings/papers/v40/Arora15.html

[13] B. Barak, J. A. Kelner, and D. Steurer, "Dictionary learning and tensor decomposition via the sum-of-squares method," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: ACM, 2015, pp. 143–151. [Online]. Available: http://doi.acm.org/10.1145/2746539.2746605

[14] T. Ma, J. Shi, and D. Steurer, "Polynomial-time tensor decompositions with sum-of-squares," in *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*. IEEE, 2016, pp. 438–446.

[15] A. Blum and J. Spencer, "Coloring random and semi-random k-colorable graphs," *J. Algorithms*, vol. 19, pp. 204–234, September 1995. [Online]. Available: http://dx.doi.org/10.1006/jagm.1995.1034

[16] U. Feige and J. Kilian, "Heuristics for finding large independent sets, with applications to coloring semi-random graphs," in *Foundations of Computer Science, 1998. Proceedings.39th Annual Symposium on*, nov 1998, pp. 674 –683.

[17] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theor.*, vol. 56, no. 5, pp. 2053–2080, May 2010. [Online]. Available: http://dx.doi.org/10.1109/TIT.2010.2044061

[18] I. Goodfellow, A. Courville, and Y. Bengio, "Large-scale feature learning with spike-and-slab sparse coding," *arXiv preprint arXiv:1206.6407*, 2012.

[19] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec 2008. [Online]. Available: https://doi.org/10.1007/s00365-007-9003-x

[20] R. O'Donnell, *Analysis of Boolean Functions*. New York, NY, USA: Cambridge University Press, 2014.

[21] D. L. Hanson and F. T. Wright, "A bound on tail probabilities for quadratic forms in independent random variables," *Ann. Math. Statist.*, vol. 42, no. 3, pp. 1079–1083, 06 1971. [Online]. Available: https://doi.org/10.1214/aoms/1177693335

[22] R. Lataa, "Estimates of moments and tails of gaussian chaoses," *Ann. Probab.*, vol. 34, no. 6, pp. 2315–2331, 11 2006. [Online]. Available: https://doi.org/10.1214/009117906000000421

[23] R. Adamczak and P. Wolff, "Concentration inequalities for non-lipschitz functions with bounded derivatives of higher order," *Probability Theory and Related Fields*, vol. 162, no. 3, pp. 531–586, Aug 2015.

[24] J. H. Kim and V. H. Vu, "Concentration of multivariate polynomials and its applications," *Combinatorica*, vol. 20, no. 3, pp. 417–434, Mar 2000. [Online]. Available: https://doi.org/10.1007/s004930070014

[25] W. Schudy and M. Sviridenko, "Concentration and moment inequalities for polynomials of independent random variables," in *SODA*, 2012.

[26] K. Luh and V. Vu, "Random matrices: l1 concentration and dictionary learning with few samples," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 2015, pp. 1409–1425.

[27] J. Błasiok and J. Nelson, "An improved analysis of the er-spud dictionary learning algorithm," *arXiv preprint arXiv:1602.05719*, 2016.

[28] S. Arora, A. Bhaskara, R. Ge, and T. Ma, "More algorithms for provable dictionary learning," *arXiv preprint arXiv:1401.0579*, 2014.

[29] P. K. Kothari and D. Steurer, "Outlier-robust moment-estimation via sum-of-squares," *arXiv preprint arXiv:1711.11581*, 2017.

[30] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon, "Learning sparsely used overcomplete dictionaries via alternating minimization," *CoRR*, vol. abs/1310.7991, 2013. [Online]. Available: http://arxiv.org/abs/1310.7991

[31] M. Aharon, M. Elad, and A. M. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *Linear algebra and its applications*, vol. 416, no. 1, pp. 48–67, 2006.

[32] P. Georgiev, F. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," *IEEE transactions on neural networks*, vol. 16, no. 4, pp. 992–996, 2005.

[33] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287 – 314, 1994, higher Order Statistics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165168494900299

[34] A. M. Frieze, M. Jerrum, and R. Kannan, "Learning linear transformations," in *FOCS*, 1996.

[35] N. Goyal, S. Vempala, and Y. Xiao, "Fourier PCA and robust tensor decomposition," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 584–593. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591875

[36] L. De Lathauwer, J. Castaing, and J. Cardoso, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Trans. on Signal Processing*, vol. 55, no. 6, pp. 2965–2973, 2007.

[37] A. Kolla, K. Makarychev, and Y. Makarychev, "How to play unique games against a semi-random adversary," in *Proceedings of 52nd IEEE symposium on Foundations of Computer Science*, ser. FOCS '11, 2011.

[38] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan, "Approximation algorithms for semi-random partitioning problems," in *Proceedings of the 44th Symposium on Theory of Computing (STOC)*. ACM, 2012, pp. 367–384.

[39] ——, "Constant factor approximations for balanced cut in the random pie model," in *Proceedings of the 46th Symposium on Theory of Computing (STOC)*. ACM, 2014.

[40] A. Perry and A. S. Wein, "A semidefinite program for unbalanced multisection in the stochastic block model," in *2017 International Conference on Sampling Theory and Applications (SampTA)*, July 2017, pp. 64–67.

[41] A. Moitra, W. Perry, and A. S. Wein, "How robust are reconstruction thresholds for community detection," *CoRR*, vol. abs/1511.01473, 2015.

[42] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan, "Learning communities in the presence of errors," *Proceedings of the Conference on Learning Theory (COLT)*, 2016.

[43] C. Mathieu and W. Schudy, "Correlation clustering with noisy input," in *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '10. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2010, pp. 712–728. [Online]. Available: http://dl.acm.org/citation.cfm?id=1873601.1873659

[44] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan, "Correlation clustering with noisy partial information," *Proceedings of the Conference on Learning Theory (COLT)*, 2015.

[45] ——, "Sorting noisy data with partial information," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. ACM, 2013, pp. 515–528.

[46] R. David and U. Feige, "On the effect of randomness on planted 3-coloring models," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2016. New York, NY, USA: ACM, 2016, pp. 77–90. [Online]. Available: http://doi.acm.org/10.1145/2897518.2897561

[47] P. Awasthi and A. Vijayaraghavan, "Clustering semi-random mixtures of gaussians," *CoRR*, vol. abs/1711.08841, 2017. [Online]. Available: http://arxiv.org/abs/1711.08841

[48] ——, "Towards learning sparsely used dictionaries with arbitrary supports," *CoRR*, vol. abs/1804.08603, 2018. [Online]. Available: http://arxiv.org/abs/1804.08603