JMLR: Workshop and Conference Proceedings vol 35:1–28, 2014

New Algorithms for Learning Incoherent and Overcomplete Dictionaries

Sanjeev Arora ARORA@CS.PRINCETON.EDU
Princeton University, Computer Science Department and Center for Computational Intractability.

Rong Ge RONGGE@MICROSOFT.COM
Microsoft Research, New England[*]

Ankur Moitra MOITRA@MIT.EDU
Massachusetts Institute of Technology, Department of Mathematics and CSAIL [†]

Abstract

In *sparse recovery* we are given a matrix $A \in \mathbb{R}^{n \times m}$ ("the dictionary") and a vector of the form AX where X is *sparse*, and the goal is to recover X. This is a central notion in signal processing, statistics and machine learning. But in applications such as *sparse coding*, edge detection, compression and super resolution, the dictionary A is unknown and has to be learned from random examples of the form $Y = AX$ where X is drawn from an appropriate distribution — this is the *dictionary learning* problem. In most settings, A is *overcomplete*: it has more columns than rows. This paper presents a polynomial-time algorithm for learning overcomplete dictionaries; the only previously known algorithm with provable guarantees is the recent work of Spielman et al. (2012) who gave an algorithm for the undercomplete case, which is rarely the case in applications. Our algorithm applies to *incoherent* dictionaries which have been a central object of study since they were introduced in seminal work of Donoho and Huo (1999). In particular, a dictionary is μ-incoherent if each pair of columns has inner product at most $\mu/\sqrt{n}$.

The algorithm makes natural stochastic assumptions about the unknown sparse vector X, which can contain $k \leq c \min(\sqrt{n}/\mu \log n, m^{1/2-\eta})$ non-zero entries (for any $\eta > 0$). This is close to the best k allowable by the best sparse recovery algorithms *even if one knows the dictionary A exactly*. Moreover, both the running time and sample complexity depend on $\log 1/\epsilon$, where ϵ is the target accuracy, and so our algorithms converge very quickly to the true dictionary. Our algorithm can also tolerate substantial amounts of noise provided it is incoherent with respect to the dictionary (e.g., Gaussian). In the noisy setting, our running time and sample complexity depend polynomially on $1/\epsilon$, and this is necessary.

1. Introduction

Finding *sparse* representations for data —signals, images, natural language— is a major focus of computational harmonic analysis (Elad, 2010; Mallat, 1998). This requires having the right dictionary $A \in \mathbb{R}^{n \times m}$ for the dataset, which allows each data point to be written as a sparse linear combination of the columns of A. For images, popular choices for the dictionary include sinusoids, wavelets, ridgelets, curvelets, etc. (Mallat, 1998) and each one is useful for different types of

[†] Part of this work was done while the author was a graduate student at Princeton University and was supported in part by NSF grants CCF-0832797, CCF-1117309, CCF-1302518, DMS-1317308, and Simons Investigator

[†] Part of this work was done while the author was a postdoc at the Institute for Advanced Study and was supported in part by NSF grant No. DMS-0835373 and by an NSF Computing and Innovation Fellowship.

features: wavelets for impulsive events, ridgelets for discontinuities in edges, curvelets for smooth curves, etc. It is common to combine such hand-designed bases into a single dictionary, which is "redundant" or "overcomplete" because $m \gg n$. This can allow sparse representation even if an image contains many different "types" of features jumbled together. In machine learning dictionaries are also used for *feature selection* (Pontil et al., 2007) and for building classifiers on top of sparse coding primitives (Kavukcuoglu et al., 2008).

In many settings hand-designed dictionaries do not do as well as dictionaries that are fit to the dataset using automated methods. In image processing such discovered dictionaries are used to perform denoising (Elad and Aharon, 2006), edge-detection (Mairal et al., 2008), super-resolution (Yang et al., 2008) and compression. The problem of discovering the best dictionary to a dataset is called *dictionary learning* and also referred to as *sparse coding* in machine learning. Dictionary learning is also a basic building block in the design of deep learning systems (Ranzato et al., 2007). See Aharon (2006); Elad (2010) for further applications. In fact, the dictionary learning problem was identified by Olshausen and Field (1997) as part of a study on internal image representations in the visual cortex. Their work suggested that basis vectors in learned dictionaries often correspond to well-known image filters such as Gabor filters.

Our goal is to design an algorithm for this problem with provable guarantees in the same spirit as recent work on nonnegative matrix factorization (Arora et al., 2012a), topic models (Arora et al., 2012b; Anandkumar et al., 2012) and mixtures models (Moitra and Valiant, 2010; Belkin and Sinha, 2010). (We will later discuss why current algorithms in Lewicki and Sejnowski (2000), Engan et al. (1999), Aharon et al. (2006), Kreutz-Delgado et al. (2003), Lee et al. (2006) do not come with such guarantees.) Designing such algorithms for dictionary learning has proved challenging. Even if the dictionary is completely known, it can be NP-hard to represent a vector u as a sparse linear combination of the columns of A (Davis et al., 1997). However for many natural types of dictionaries, the problem of finding a sparse representation is computationally easy. The pioneering work of Donoho and Huo (1999), Donoho and Elad (2003) and Gribonval and Nielsen (2003) (building on the uncertainty principle of Donoho and Stark (1999)) presented a number of important examples (in fact, the ones we used above) of dictionaries that are *incoherent* and showed that ℓ_1-minimization can find a sparse representation in a known, incoherent dictionary if one exists.

Definition 1 (μ-incoherent) *An $n \times m$ matrix A whose columns are* unit *vectors is μ*-incoherent *if $\forall i \neq j$ we have $\langle A_i, A_j \rangle \leq \mu/\sqrt{n}$. We will refer to A as* incoherent *if μ is $O(\log n)$.*

A randomly chosen dictionary is incoherent with high probability (even if $m = n^{100}$). Donoho and Huo (1999) gave many other important examples of incoherent dictionaries, such as one constructed from *spikes* and *sines*, as well as those built up from wavelets and sines, or even wavelets and ridgelets. There is a rich body of literature devoted to incoherent dictionaries (see additional references in Gilbert et al. (2003)). Donoho and Huo (1999) proved that given $u = Av$ where v has k nonzero entries, where $k \leq \sqrt{n}/2\mu$, *basis pursuit* (solvable by a linear program) recovers v exactly and it is unique. Gilbert et al. (2003) (and subsequently Tropp et al. (2003)) gave algorithms for recovering v even in the presence of additive noise. Tropp (2004) gave a more general *exact recovery condition* (ERC) under which the sparse recovery problem for incoherent dictionaries can be algorithmically solved. All of these require $n > k^2\mu^2$. In a foundational work, Candes et al. (2006) showed that basis pursuit solves the sparse recovery problem even for $n = O(k \log(m/k))$ if A satisfies the weaker *restricted isometry property* (Candes and Tao, 2005). Also if A is a full-rank square matrix, then we can compute v from $A^{-1}u$, trivially. But our focus here will be on incoherent and overcomplete dictionaries; extending these results to RIP matrices is left as a major open problem.

The main result in this paper is an algorithm that *provably* learns an unknown, incoherent dictionary from random samples $Y = AX$ where X is a vector with at most $k \le c \min(\sqrt{n}/\mu \log n, m^{1/2-\eta})$ non-zero entries (for any $\eta > 0$, small enough constant $c > 0$ depending on η). Hence we can allow almost as many non-zeros in the hidden vector X as the best sparse recovery algorithms which *assume that the dictionary A is known*. The precise requirements that we place on the distributional model are described in Section 1.2. We can relax some of these conditions at the cost of increased running time or requiring X to be more sparse. Finally, our algorithm can tolerate a substantial amount of additive noise, an important consideration in most applications including sparse coding, provided it is independent and uncorrelated with the dictionary.

1.1. Related Works

Algorithms used in practice Dictionary learning is solved in practice by variants of *alternating minimization*. Lewicki and Sejnowski (2000) gave the first approach; subsequent popular approaches include the *method of optimal directions* (MOD) of Engan et al. (1999), and *K-SVD* of Aharon et al. (2006). The general idea is to maintain a guess for A and X and at every step either update X (using basis pursuit) or update A by, say, solving a least squares problem. Provable guarantees for such algorithms have proved difficult because the initial guesses may be very far from the true dictionary, causing basis pursuit to behave erratically. Also, the algorithms could converge to a dictionary that is not incoherent, and thus unusable for sparse recovery. (In practice, these heuristics do often work.)

Algorithms with guarantees An elegant paper of Spielman et al. (2012) shows how to provably recover A *exactly* if it has full column rank, and X has at most $\sqrt{n}$ nonzeros. However, requiring A to be full column rank precludes most interesting applications where the dictionary is redundant and hence cannot have full column rank (see Donoho and Huo (1999); Elad (2010); Mallat (1998)). Moreover, the algorithm in Spielman et al. (2012) is not noise tolerant.

After the initial announcement of this work, Agarwal et al. (2013b,a) independently gave provable algorithms for learning overcomplete and incoherent dictionaries. Their first paper (Agarwal et al., 2013b) requires the entries in X to be independent random ± 1 variables. Their second (Agarwal et al., 2013a) gives an algorithm –a version of alternating minimization– that converges to the correct dictionary given a good initial dictionary (such a good initialization can only be found using Agarwal et al. (2013b) in special cases, or more generally using this paper). Unlike our algorithms, theirs assume the sparsity of X is at most $n^{1/4}$ or $n^{1/6}$ (assumption A4 in both papers), which are far from the $n^{1/2}$ limit of incoherent dictionaries. The main change from the initial version of our paper is that we have improved the dependence of our algorithms from $\text{poly}(1/\epsilon)$ to $\log 1/\epsilon$ (see Section 4).

After this work, Barak et al. (2014) give an quasi-polynomial time algorithm for dictionary learning using sum-of-squares SDP hierarchy. The algorithm can output an approximate dictionary even when sparsity is almost linear in the dimensions with weaker assumptions.

Independent Component Analysis When the entries of X are independent, algorithms for *independent component analysis or ICA* (Comon, 1994) can recover A. Frieze et al. (1996) gave a provable algorithm that recovers A up to arbitrary accuracy, provided entries in X are non-Gaussian (when X is Gaussian, A is only determined up to rotations anyway). Subsequent works considered the overcomplete case and gave provable algorithms even when A is $n \times m$ with $m > n$ (Lathauwer et al., 2007; Goyal et al., 2014).

However, these algorithms are incomparable to ours since the algorithms are relying on different assumptions (independence vs. sparsity). With sparsity assumption, we can make much weaker assumptions on how X is generated. In particular, all these algorithms require the support Ω of the vector X to be at least 3-wise independent ($\mathbf{Pr}[u, v, w \in \Omega] = \mathbf{Pr}[u \in \Omega]\,\mathbf{Pr}[v \in \Omega]\,\mathbf{Pr}[w \in \Omega]$) in the undercomplete case and 4-wise independence in the overcomplete case . Our algorithm only requires the support S to have bounded moments ($\mathbf{Pr}[u, v, w \in \Omega] \le \Lambda\,\mathbf{Pr}[u \in \Omega]\,\mathbf{Pr}[v \in \Omega]\,\mathbf{Pr}[w \in \Omega]$ where Λ is a large constant or even a polynomial depending on m, n, k, see Definition 5). Also, because our algorithm relies on the sparsity constraint, we are able to get almost exact recover in the noiseless case (see Theorem 4 and Section 4). This kind of guarantee is impossible for ICA without sparsity assumption.

1.2. Our Results

A range of results are possible which trade off more assumptions with better performance. We give two illustrative ones: the first makes the most assumptions but has the best performance; the second has the weakest assumptions and somewhat worse performance. The theorem statements will be cleaner if we use asymptotic notation: the parameters k, n, m will go to infinity and the constants denoted as "$O(1)$" are arbitrary so long as they do not grow with these parameters.

First we define the class of distributions that the k-sparse vectors must be drawn from. We will be interested in distributions on k-sparse vectors in $\mathbb{R}^m$ where each coordinate is nonzero with probability $\Theta(k/m)$ (the constant in $\Theta(\cdot)$ can differ among coordinates).

Definition 2 (Distribution class Γ and its moments) *The distribution is in class Γ if (i) each nonzero X_i has expectation 0 and lies in $[-C, -1] \cup [1, C]$ where $C = O(1)$. (ii) Conditioned on any subset of coordinates in X being nonzero, the values X_i are independent of each other.*

The distribution has bounded ℓ-wise moments *if the probability that X is nonzero in any subset S of ℓ coordinates is at most c^ℓ times $\prod_{i \in S} \mathbf{Pr}[X_i \neq 0]$ where $c = O(1)$.*

Remark: (i) The bounded moments condition trivially holds for any constant ℓ if the set of nonzero locations is a random subset of size k. The *values* of these nonzero locations are allowed to be distributed very differently from one another. (ii) The requirement that nonzero X_i's be bounded away from zero in magnitude is similar in spirit to the *Spike-and-Slab Sparse Coding* (S3C) model of Goodfellow et al. (2012), which also encourages nonzero latent variables to be bounded away from zero to avoid degeneracy issues that arise when some coefficients are much larger than others. (iii) In the rest of the paper we will be focusing on the case when $C = 1$, all the proofs generalize directly to the case $C > 1$ by losing constant factors in the guarantees.

Because of symmetry in the problem, we can only hope to learn dictionary A up to permutation and sign-flips. We say two dictionaries are column-wise ϵ-close, if after appropriate permutation and flipping the corresponding columns are within distance ϵ.

Definition 3 *Two dictionaries $A, B \in \mathbb{R}^{n \times m}$ are column-wise ϵ-close, if there exists a permutation π and $\theta \in \{\pm 1\}^m$ such that $\|(A_i) - \theta_i (B)_{\pi(i)}\| \le \epsilon$.*

Later when we are talking about two dictionaries that are ϵ-close, we always assume the columns are ordered correctly so that $\|A_i - B_i\| \le \epsilon$.

Theorem 4 *There is a polynomial time algorithm to learn a μ-incoherent dictionary A from random examples. With high probability the algorithm returns a dictionary $\hat{A}$ that is column-wise ϵ close to A given random samples of the form $Y = AX$, where $X \in \mathbb{R}^n$ is chosen according to some distribution in Γ and A is in $\mathbb{R}^{n\times m}$:*

- *If $k \le c\min(m^{2/5}, \frac{\sqrt{n}}{\mu\log n})$ and the distribution has bounded 3-wise moments, $c > 0$ is a universal constant, then the algorithm requires p_1 samples and runs in time $\widetilde{O}(p_1^2 n)$.*
- *If $k \le c\min(m^{(\ell-1)/(2\ell-1)}, \frac{\sqrt{n}}{\mu\log n})$ and the distribution has bounded ℓ-wise moments, $c > 0$ is a constant only depending on ℓ, then the algorithm requires p_2 samples and runs in time $\widetilde{O}(p_2^2 n)$*
- *Even if each sample is of the form $Y^{(i)} = AX^{(i)} + \eta_i$, where η_i's are independent spherical Gaussian noise with standard deviation $\sigma = o(\sqrt{n})$, the algorithms above still succeed provided the number of samples is at least p_3 and p_4 respectively.*

In particular $p_1 = \Omega((m^2/k^2)\log m + mk^2\log m + m\log m\log 1/\epsilon)$ and $p_2 = \Omega((m/k)^{\ell-1}\log m + mk^2\log m\log 1/\epsilon)$ and p_3 and p_4 are larger by a σ^2/ϵ^2 factor.

Remark: The sparsity that our algorithm can tolerate – the minimum of $\frac{\sqrt{n}}{\mu\log n}$ and $m^{1/2-\eta}$ – approaches the sparsity that the best known algorithms require *even if A is known*.
Although the running time and sample complexity of the algorithm are relatively large polynomials, there are many ways to optimize the algorithm. See the discussion in Section 5.

Now we describe the other result which requires fewer assumptions on how the samples are generated, but require more stringent bounds on the sparsity:

Definition 5 (Distribution class $\mathcal{D}$) *A distribution is in class $\mathcal{D}$ if (i) the events $X_i \ne 0$ have* weakly bounded *second and third moments, in the sense that $\mathbf{Pr}[X_i \ne 0$ and $X_j \ne 0] \le n^{\epsilon}\,\mathbf{Pr}[X_i \ne 0]\,\mathbf{Pr}[X_j \ne 0]$, $\mathbf{Pr}[X_i, X_j, X_t \ne 0] \le o(n^{1/4})\,\mathbf{Pr}[X_i \ne 0]\,\mathbf{Pr}[X_j \ne 0]\,\mathbf{Pr}[X_t \ne 0]$. (ii) Each nonzero X_i is in $[-C,-1]\cup[1,C]$ where $C = O(1)$.*

The following theorem is proved similarly to Theorem 4, and is sketched in Appendix G.

Theorem 6 *There is a polynomial time algorithm to learn a μ-incoherent dictionary A from random examples of the form $Y = AX$, where X is chosen according to some distribution in $\mathcal{D}$. If $k \le c\min(m^{1/4}, \frac{n^{1/4-\epsilon/2}}{\sqrt{\mu}})$ and we are given $p \ge \Omega(\max(m^2/k^2\log m, \frac{mn^{3/2}\log m\log n}{k^2\mu}))$ samples , then the algorithm succeeds with high probability, and the output dictionary is column-wise $\epsilon = O(k\sqrt{\mu}/n^{1/4-\epsilon/2})$ close to the true dictionary. The algorithm runs in time $\tilde{O}(p^2 n + m^2 p)$. The algorithm is also noise-tolerant as in Theorem 4.*

1.3. Proof Outline

The key observation in the algorithm is that we can test whether two samples share the same dictionary element (see Section 2.1). Given this information, we can build a graph whose vertices are the samples, and edges correspond to samples that share the same dictionary element. A large cluster in this graph corresponds to the set of all samples with $X_i \neq 0$. In Section 2.2 we give an algorithm for finding all the large clusters. Then we show how to recover the dictionary given the clusters in Section 3. This allows us to get a rough estimate of the dictionary matrix. Section 4 gives an algorithm for refining the solution in the noiseless case. The three main parts of the techniques are:

Overlapping Clustering: Heuristics such as MOD (Engan et al., 1999) or K-SVD (Aharon et al., 2006) have a cyclic dependence: If we knew A, we could solve for X and if we knew all of the X's we could solve for A. Our main idea is to break this cycle by (without knowing A) finding all of the samples where $X_i \neq 0$. We can think of this as a cluster $\mathcal{C}_i$. Although our strategy is to cluster a random graph, what is crucial is that we are looking for an *overlapping* clustering since each sample X belongs to k clusters! Many of the algorithms which have been designed for finding overlapping clusterings (e.g. Arora et al. (2012c), Balcan et al. (2013)) have a poor dependence on the maximum number of clusters that a node can belong to. Instead, we give a simple combinatorial algorithm based on triplet (or higher-order) tests that recovers the underlying, overlapping clustering. In order to prove correctness of our combinatorial algorithm, we rely on tools from discrete geometry, namely the *piercing number* (Matousek, 2002; Alon and Kleitman, 1992).

Recovering the Dictionary: Next, we observe that there are a number of natural algorithms for recovering the dictionary once we know the clusters $\mathcal{C}_i$. We can think of a random sample from $\mathcal{C}_i$ as applying a filter to the samples we are given, and filtering out only those samples where $X_i \neq 0$. The claim is that this distribution will have a much larger variance along the direction A_i than along other directions, and this allows us to recovery the dictionary either using a certain averaging algorithm, or by computing the largest singular vector of the samples in $\mathcal{C}_i$. In fact, this latter approach is similar to K-SVD (Aharon et al., 2006) and hence our analysis yields insights into why these heuristics work.

Fast Convergence: The above approach yields provable algorithms for dictionary learning whose running time and sample complexity depend polynomially on $1/\epsilon$. However once we have a suitably good approximation to the true dictionary, can we converge at a much faster rate? We analyze a simple alternating minimization algorithm ITERATIVE AVERAGE and we derive a formula for its updates where we can analyze it by thinking of it instead as a noisy version of the matrix power method (see Lemma 20). This analysis is inspired by recent work on analyzing alternating minimization for the matrix completion problem (Jain et al., 2013; Hardt, 2013), and we obtain algorithms whose running time and sample complexity depends on $\log 1/\epsilon$. Hence we get algorithms that converge rapidly to the true dictionary while simultaneously being able to handle almost the same sparsity as in the sparse recovery problem where A is known!

NOTATION: Throughout this paper, we will use $Y^{(i)}$ to denote the i^{th} sample and $X^{(i)}$ as the vector that generated it – i.e. $Y^{(i)} = AX^{(i)}$. Let $\Omega^{(i)}$ denote the support of $X^{(i)}$. For a vector X let X_i be the i^{th} coordinate. For a matrix $A \in \mathbb{R}^{n \times m}$ (especially the dictionary matrix), we use A_i to denote the i-th column (the i-th dictionary element). Also, for a set $S \subset \{1, 2, ..., m\}$, we use A_S to denote the submatrix of A with columns in S. We will use $\|A\|_F$ to denote the Frobenius norm and $\|A\|$ to denote the spectral norm. Moreover we will use Γ to denote the distribution on k-sparse vectors X that is used to generate our samples, and Γ_i will denote the restriction of this distribution to vectors X where $X_i \neq 0$. When we are working with a graph G we will use $\Gamma_G(u)$ to denote the set of neighbors of u in G. Throughout the paper "with high probability" means the probability is at least $1 - n^{-\Delta}$ for large enough Δ.

2. The Connection Graph and Overlapping Clustering

2.1. The Connection Graph

In this part we show how to test whether two samples share the same dictionary element, i.e., whether the supports $\Omega^{(i)}$ and $\Omega^{(j)}$ intersect. The idea is we can check the inner-product of $Y^{(i)}$ and $Y^{(j)}$, which can be decomposed into the sum of inner-products of dictionary elements

$$\langle Y^{(i)}, Y^{(j)} \rangle = \sum_{p \in \Omega^{(i)}, q \in \Omega^{(j)}} \langle A_p, A_q \rangle X_p^{(i)} X_q^{(j)}$$

If the supports are disjoint, then each of the terms above is small since $\langle A_p, A_q \rangle \leq \mu/\sqrt{n}$ by the incoherence assumption. Moreover we will use the Hanson-Wright inequality (see Appendix A) to show that each sum is close to its expectation. This observation is formalized in the following lemma:

Lemma 7 *Suppose $k\mu < \frac{\sqrt{n}}{C' \log n}$ for large enough constant C' (depending on C in Definition 2). Then if $\Omega^{(i)}$ and $\Omega^{(j)}$ are disjoint, with high probability $|\langle Y^{(i)}, Y^{(j)} \rangle| < 1/2$.*

We defer the proof of this lemma to Appendix A. In our algorithm, we build the following graph:

Definition 8 *Given p samples $Y^{(1)}, Y^{(2)}, ..., Y^{(p)}$, build a* connection graph *on p nodes where i and j are connected by an edge if and only if $|\langle Y^{(i)}, Y^{(j)} \rangle| > 1/2$.*

This graph will "miss" some edges, since if a pair $X^{(i)}$ and $X^{(j)}$ have intersecting support we do not necessarily meet the above condition. But by Lemma 7 (with high probability) this graph will not have any false positives:

Corollary 9 *With high probability, each edge (i, j) present in the connection graph corresponds to a pair where $\Omega^{(i)}$ and $\Omega^{(j)}$ have non-empty intersection.*

Consider a sample $Y^{(1)}$ for which there is an edge to both $Y^{(2)}$ and $Y^{(3)}$. This means that there is some coordinate i in both $\Omega^{(1)}$ and $\Omega^{(2)}$ and some coordinate i' in both $\Omega^{(1)}$ and $\Omega^{(3)}$. However the challenge is that we do not immediately know if $\Omega^{(1)}, \Omega^{(2)}$ and $\Omega^{(3)}$ have a common intersection or not.

2.2. Overlapping Clustering

Our goal in this section is to determine which samples Y have $X_i \neq 0$ just from the connection graph. To do this, we will identify a combinatorial condition that allows us to decide whether or not a set of three samples $Y^{(1)}, Y^{(2)}$ and $Y^{(3)}$ that have supports $\Omega^{(1)}, \Omega^{(2)}$ and $\Omega^{(3)}$ respectively – have a common intersection or not. From this condition, it is straightforward to give an algorithm that correctly groups together all of the samples Y that have $X_i \neq 0$. In order to reduce the number of letters used we will focus on the first three samples $Y^{(1)}, Y^{(2)}$ and $Y^{(3)}$ although all the claims and lemmas hold for all triples.

Suppose we are given two samples $Y^{(1)}$ and $Y^{(2)}$ with supports $\Omega^{(1)}$ and $\Omega^{(2)}$ where $\Omega^{(1)} \cap \Omega^{(2)} = \{i\}$. We will prove that this pair can be used to recover all the samples Y for which $X_i \neq 0$. This will follow because we will show that the expected number of common neighbors between $Y^{(1)}, Y^{(2)}$ and Y will be large if $X_i \neq 0$ and otherwise will be small. So throughout this subsection let us consider a sample $Y = AX$ and let Ω be its support. We will need the following elementary claim, whose proof we defer to Appendix A.

Claim 10 *Suppose $\Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)} \neq \emptyset$, then $Pr_Y[$for all $j = 1, 2, 3, |\langle Y, Y^{(j)} \rangle| > 1/2] \geq \frac{k}{2m}$*

This claim establishes a lower bound on the expected number of common neighbors of a triple, if they have a common intersection. Next we establish an upper bound, if they don't have a common intersection. Suppose $\Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)} = \emptyset$. In principle we should be concerned that Ω could still intersect each of $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$ in different locations. Let $a = |\Omega^{(1)} \cap \Omega^{(2)}|$, $b = |\Omega^{(1)} \cap \Omega^{(3)}|$ and $c = |\Omega^{(2)} \cap \Omega^{(3)}|$. We defer the proof of the following lemma to Appendix A:

Lemma 11 *Suppose that $\Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)} = \emptyset$. Then the probability that Ω intersects each of $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$ is at most*

$$\frac{k^6}{m^3} + \frac{3k^3(a+b+c)}{m^2}$$

Note that if Γ has bounded higher order moment, the probability that two sets of size k intersect in at least Q elements is at most $(\frac{k^2}{m})^Q$. Hence we can assume that with high probability there is *no* pair of samples whose supports intersect in more than a constant number of locations. When Γ only has bounded 3-wise moment see Appendix A.1.

Let us quantitatively compare our lower and upper bound: If $k \leq cm^{2/5}$ then the expected number of common neighbors for a triple with $\Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)} \neq \emptyset$ is much larger than the expected number of common neighbors of a triple whose common intersection is empty. Under this condition, if we take $p = O(m^2/k^2 \log n)$ samples each triple with a common intersection will have at least T common neighbors, and each triple whose common intersection is empty will have less than $T/2$ common neighbors.

Hence we can search for a triple with a common intersection as follows: We can find a pair of samples $Y^{(1)}$ and $Y^{(2)}$ whose supports intersect. We can take a neighbor $Y^{(3)}$ of $Y^{(1)}$ in the connection graph (at random), and by counting the number of common neighbors of $Y^{(1)}$, $Y^{(2)}$ and $Y^{(3)}$ we can decide whether or not their supports have a common intersection.

Definition 12 *We will call a pair of samples $Y^{(1)}$ and $Y^{(2)}$ an* identifying pair *for coordinate i if the intersection of $\Omega^{(1)}$ and $\Omega^{(2)}$ is exactly $\{i\}$.*

Theorem 13 *The output of* OVERLAPPINGCLUSTER *is an overlapping clustering where each set corresponds to some i and contains all $Y^{(j)}$ for which $i \in \Omega^{(j)}$. The algorithm runs in time $\widetilde{O}(p^2 n)$ and succeeds with high probability if $k \leq c \min(m^{2/5}, \frac{\sqrt{n}}{\mu \log n})$ and if $p = \Omega(\frac{m^2 \log m}{k^2})$*

Proof: We can use Lemma 7 to conclude that each edge in G corresponds to a pair whose support intersects. We can appeal to Lemma 11 and Claim 10 to conclude that for $p = \Omega(m^2/k^2 \log m)$,

Algorithm 1 OVERLAPPINGCLUSTER, **Input:** p samples $Y^{(1)}, Y^{(2)}, ..., Y^{(p)}$

```
1. Compute a graph $G$ on $p$ nodes where there is an edge between $i$ and $j$ iff $|\langle Y^{(i)}, Y^{(j)}\rangle| > 1/2$
2. Set $T = \frac{pk}{10m}$
3. Repeat $\Omega(m \log^2 m)$ times:
4.     Choose a random edge $(u, v)$ in $G$
5.     Set $S_{u,v} = \{w : |\Gamma_G(u) \cap \Gamma_G(v) \cap \Gamma_G(w)| \geq T\} \cup \{u, v\}$
6. Delete any set $S_{u,v}$ where $u, v$ are contained in a strictly smaller set $S_{a,b}$ (also delete any duplicates)
7. Output the remaining sets $S_{u,v}$
```

with high probability each triple with a common intersection has at least T common neighbors, and each triple without a common intersection has at most $T/2$ common neighbors.

In fact, for a random edge $(Y^{(1)}, Y^{(2)})$, the probability that the common intersection of $\Omega^{(1)}$ and $\Omega^{(2)}$ is exactly $\{i\}$ is $\Omega(1/m)$ because we know that they do intersect, and that intersection has a constant probability of being size one and it is uniformly distributed over m possible locations. Appealing to a coupon collector argument we conclude that if the inner loop is run at least $\Omega(m \log^2 m)$ times then the algorithm finds an identifying pair (u, v) for each column A_i with high probability.

Note that we may have pairs that are not an identifying triple for some coordinate i. However, any other pair (u, v) found by the algorithm must have a common intersection. Consider for example a pair (u, v) where u and v have a common intersection $\{i, j\}$. Then we know that there is some other pair (a, b) which is an identifying pair for i and hence $S_{a,b} \subset S_{u,v}$. (In fact this containment is strict, since $S_{u,v}$ will also contain a set corresponding to an identifying pair for j too). Hence the second-to-last step in the algorithm will necessarily delete all such non-identifying pairs $S_{u,v}$.

What is the running time of this algorithm? We need $O(p^2 n)$ time to build the connection graph, and the loop takes $\widetilde{O}(pmn)$ time. Finally, the deletion step requires time $\widetilde{O}(m^2)$ since there will be $\widetilde{O}(m)$ pairs found in the previous step and for each pair of pairs, we can delete $S_{u,v}$ if and only if there is a strictly smaller $S_{a,b}$ that contains u and v. This concludes the proof of correctness of the algorithm, and its running time analysis. ■

3. Recovering the Dictionary

3.1. Finding the Relative Signs

Here we show how to recover the column A_i once we have learned which samples Y have $X_i \neq 0$. We will refer to this set of samples as the "cluster" $\mathcal{C}_i$. The key observation is that if $\Omega^{(1)}$ and $\Omega^{(2)}$ uniquely intersect in index i then the sign of $\langle Y^{(1)}, Y^{(2)}\rangle$ is equal to the sign of $X_i^{(1)} X_i^{(2)}$. If there are enough such pairs $Y^{(1)}$ and $Y^{(2)}$, we can determine not only which samples Y have $X_i \neq 0$ but also which pairs of samples Y and Y' have $X_i, X'_i \neq 0$ and $\text{sign}(X_i) = \text{sign}(X'_i)$. This is the main step of the algorithm OVERLAPPINGAVERAGE, and we defer its description and the proof of the following theorem to Appendix B.

Theorem 14 *If the input to* OVERLAPPINGAVERAGE $\mathcal{C}_1, ..., \mathcal{C}_m$ *are the true clusters* $\{j : i \in \Omega^{(j)}\}$ *up to permutation, then the algorithm outputs a dictionary* $\hat{A}$ *that is column-wise* ϵ*-close to* A *with high probability if* $k \le \min(\sqrt{m}, \frac{\sqrt{n}}{\mu})$ *and if* $p = \Omega\left(\max(m^2 \log m/k^2, m \log m/\epsilon^2)\right)$ *Furthermore the algorithm runs in time* $O(p^2)$.

3.2. An Approach via SVD

In Appendix C we give an alternative algorithm for recovering the dictionary based instead on SVD. Intuitively if we take all the samples whose support contains index j, then every such sample $Y^{(i)}$ has a component along direction A_j. Therefore direction A_j should have the largest variance and can be found by SVD. The advantage is that methods like K-SVD which are quite popular in practice also rely on finding directions of maximum variance, so the analysis we provide here yields insights into why these approaches work. However, the crucial difference is that we rely on finding the correct overlapping clustering in the first step of our dictionary learning algorithms, whereas K-SVD and approaches like approximate it via their current guess for the dictionary.

3.3. Noise Tolerance

Here we elaborate on why the algorithm can tolerate noise provided that the noise is *uncorrelated* with the dictionary (e.g. Gaussian noise). The observation is that in constructing the connection graph, we only make use of the inner products between pairs of samples $Y^{(1)}$ and $Y^{(2)}$, the value of which is roughly preserved under various noise models. In turn, the overlapping clustering is a purely combinatorial algorithm that only makes use of the connection graph. Finally, we recover the dictionary A using singular value decomposition, which is well-known to be stable under noise (e.g. Wedin's Theorem 35).

4. Refining the Solution

Earlier sections gave noise-tolerant algorithms for the dictionary learning problem with sample complexity $O(\text{poly}(n, m, k)/\epsilon^2)$. This dependency on ϵ is necessary for any noise-tolerant algorithm since even if the dictionary has only one vector, we need $O(1/\epsilon^2)$ samples to estimate the vector in presence of noise. However when Y is exactly equal to AX we can hope to recover the dictionary with better running time and much fewer samples. In particular, Geng et al. (2013) recently established that ℓ_1-minimization is locally correct for incoherent dictionaries, therefore it seems plausible that given a very good estimate for A there is some algorithm that computes a refined estimate of A whose running time and sample complexity have a better dependence on ϵ.

In this section we analyze the local-convergence of an algorithm that is similar to K-SVD (Aharon et al., 2006); see Algorithm 2 ITERATIVEAVERAGE. Recall B_S denotes the submatrix of B whose columns are indices in S; also, $P^+ = (P^T P)^{-1} P^T$ is the left-pseudoinverse of the matrix P. Hence $P^+ P = I$, PP^+ is the projection matrix to the span of columns of P.

The key lemma of this section shows the error decreases by a constant factor in each round of ITERATIVEAVERAGE (provided that it was suitably small to begin with). Let $\epsilon_0 \le 1/100k$. We will defer the proofs of some of the intermediate claims to Appendix D.

Theorem 15 *Suppose the dictionary* A *is* μ*-incoherent with* $\mu/\sqrt{n} < 1/k \log k$*, initial solution is* $\epsilon < \epsilon_0$ *close to the true solution (i.e. for all* i $\|B_i - A_i\| \le \epsilon$*). With high probability the output of* ITERATIVEAVERAGE *is a dictionary* B' *that satisfies* $\|B'_i - A_i\| \le (1 - \delta)\epsilon$*, where* δ *is a universal positive constant. Moreover, the algorithm runs in time* $O(qnk^2)$ *and succeeds with high probability when number of samples* $q = \Omega(m \log^2 m)$.

Algorithm 2 ITERATIVEAVERAGE, **Input:** Initial estimation B, $\|B_i - A_i\| \le \epsilon$, q samples (independent of B) $Y^{(1)}, Y^{(2)}, ...Y^{(q)}$

```
For each sample $i$, let $\Omega^{(i)} = \{j : |\langle Y^{(i)}, B_j\rangle| > 1/2\}$
For each dictionary element $j$
    Let $\mathcal{C}_j^+$ be the set of samples that have inner product more than $1/2$ with $B^{(j)}$ ($\mathcal{C}_j^+ = \{i : \langle Y^{(i)}, B_j\rangle > 1/2\}$)
    For each sample $i$ in $\mathcal{C}_j^+$
        Let $\hat{X}^{(i)} = B^+_{\Omega^{(i)}} Y^{(i)}$
        Let $Q_{i,j} = Y^{(i)} - \sum_{t \in \Omega^{(i)} \setminus \{j\}} B_t \hat{X}^{(i)}_t$
    Let $B'_j = \sum_{i \in \mathcal{C}_j^+} Q_{i,j} / \| \sum_{i \in \mathcal{C}_j^+} Q_{i,j} \|$.
Output $B'$.
```

We will analyze the update made to the first column B_1, and the same argument will work for all columns (and hence we can apply a union bound to complete the proof). To simplify the proof, we will let ξ denote arbitrarily small constants (whose precise value will change from line to line). First, we establish some basic claims that will be the basis for our analysis of ITERATIVEAVERAGE.

Claim 16 *Suppose A is a μ incoherent matrix with $\mu/\sqrt{n} < 1/k \log k$. If for all i, $\|B_i - A_i\| \le \epsilon_0$ then* ITERATIVEAVERAGE *recovers the correct support for each sample (i.e. $\Omega^{(i)} = supp(X^{(i)})$) and the correct sign (i.e. $\mathcal{C}_j^+ = \{j : X_j^{(i)} > 0\}$)* [1]

Claim 17 *The set of columns $\{B_i\}_i$ is $\mu' = \mu + O(k/\sqrt{n})$-incoherent where $\mu'/\sqrt{n} \le 1/10k$.*

To simplify the notation, let us permute the samples so that $\mathcal{C}_1^+ = \{1, 2, ..., l\}$. The probability that $X_1^{(i)} > 0$ is $\Theta(k/m)$ and so for $q = \Theta(m \log^2 m)$ samples with high probability the number of samples l where $X_1^{(i)} > 0$ is $\Omega(qk/m) = \Omega(k \log^2 m)$.

Definition 18 *Let M_i be the matrix $(0, B_{\Omega^{(i)} \setminus \{1\}}) B^+_{\Omega^{(i)}}$.*

Then we can write $Q_{i,1} = (I - M_i)Y^{(i)}$. Let us establish some basic properties of M_i that we will need in our analysis:

Claim 19 *M_i has the following properties: (1) $M_i B_1 = 0$ (2) For all $j \in \Omega^{(i)} \setminus \{1\}$, $M_i B_j = B_j$ and (3) $\|M_i\| \le 1 + \xi$*

Proof: The first and second property follow immediately from the definition of M_i, and the third property follows from the Gershgorin disk theorem. ■

[1] Notice that this is not a "with high probability" statement, the support is *always* correctly recovered. That is why we use $\Omega^{(i)}$ both in the algorithm and for the true support

For the time being, we will consider the vector $\hat{B}_1 = \sum_{i=1}^l Q_{i,1}/\sum_{i=1}^l X_1^{(i)}$. We cannot compute this vector directly (note that $\hat{B}_1$ and B_1' are in general different) but first we will show that $\hat{B}_1$ and A_1 are suitably close. To accomplish this, we will first find a convenient expression for the error:

Lemma 20

$$A_1 - \hat{B}_1 = \sum_{i=1}^l \frac{X_1^{(i)}}{\sum_{i=1}^l X_1^{(i)}} M_i(A_1 - B_1) - \frac{\sum_{i=1}^l \sum_{j \in \Omega^{(i)} \setminus \{1\}} (I - M_i)(A_j - B_j) X_j^{(i)}}{\sum_{i=1}^l X_1^{(i)}}. \quad (1)$$

Proof: The proof is mostly carefully reorganizing terms and using properties of M_i's to simplify the expression. See Appendix D for details. ■

We will analyze the two terms in the above equation separately. The second term is the most straightforward to bound, since it is the sum of independent vector-valued random variables (after we condition on the support $\Omega^{(i)}$ of each sample in $\mathcal{C}_1^+$.

Claim 21 *If $l > \Omega(k \log^2 m)$, then with high probability the second term of Equation (1) is bounded by $\epsilon/100$.*

All that remains is to bound the first term. Note that the coefficient of $\|M_i(A_1 - B_1)\|$ is independent of the support, and so the first term will converge to its expectation - namely $\mathbf{E}[\|M_i(A_1 - B_1)\|]$. So it suffices to bound this expectation.

Lemma 22 $\mathbf{E}[\|M_i(A_1 - B_1)\|] \le (1 - \delta)\epsilon$.

Proof: We will break up $A_1 - B_1$ onto its component (x) in the direction B_1 and its orthogonal component (y) in $B_1^\perp$. First we bound the norm of x:

$$\|x\| = |\langle A_1 - B_1, B_1 \rangle| = |\langle A_1 - B_1, A_1 - B_1 \rangle|/2 \le \epsilon^2/2$$

Next we consider the component y. Consider the supports $\Omega^{(1)}$ and $\Omega^{(2)}$ of two random samples from $\mathcal{C}_1^+$. These sets certainly intersect at least once, since both contain $\{1\}$. Yet with probability at least $2/3$ this is their only intersection (e.g. see Claim 30). If so, let $S = (\Omega^{(1)} \cup \Omega^{(2)}) \setminus \{1\}$. Recall that $\|B_S^T\| \le 1 + \xi$. However $B_S^T y$ is the concatenation of $B_{\Omega^{(1)}}^T y$ and $B_{\Omega^{(2)}}^T y$ and so we conclude that $\|B_{\Omega^{(1)}}^T y\| + \|B_{\Omega^{(2)}}^T y\| \le (1 + \xi)\sqrt{2}$. Since the spectral norm of $(0, B_{\Omega^{(i)} \setminus \{1\}})$ is bounded, we conclude that $\|M_1 y\| + \|M_2 y\| \le (1 + \xi)\sqrt{2}$. This implies that

$$\mathbf{E}[\|M_i(A_1 - B_1)\|] \le \mathbf{E}[\|M_i x\|] + \mathbf{E}[\|M_i y\|] \le (2/3)(1 + \xi)(\sqrt{2}/2)\epsilon + (1/3)(1 + \xi)\epsilon + \epsilon^2/2$$

And this is indeed at most $(1 - \delta)\epsilon$ which concludes the proof of the lemma. ■

Combining the two claims, we know that with high probability $\hat{B}_1$ has distance at most $(1 - \delta)\epsilon$ to A_1. However, B_1' is not equal to $\hat{B}_1$ (and we cannot compute $\hat{B}_1$ because we do not know the normalization factor). The key observation here is $\hat{B}_1$ is a multiple of B_1', the vector B_1' and A_1 all have unit norm, so if $\hat{B}_1$ is close to A_1 the vector B_1' must also be close to A_1.

Claim 23 *If x and y are unit vectors, and x' is a multiple of x then $\|x' - y\| \le \epsilon < 1$ implies that $\|x - y\| \le \epsilon\sqrt{1 + \epsilon^2}$*

This concludes the proof of Theorem 15. To bound the running time, observe that for each sample, the main computations involve computing the pseudo-inverse of a $n \times k$ matrix, which takes $O(nk^2)$ time.

5. Discussion

This paper shows it is possible to provably get around the chicken-and-egg problem inherent in dictionary learning: not knowing A seems to prevents recovering X's and vice versa. By using combinatorial techniques to recover the support of each X without knowing the dictionary, our algorithm suggests a new way to design algorithms.

Currently the running time is $\widetilde{O}(p^2 n)$ time, which may be too slow for large-scale problems. But our algorithm suggests more heuristic versions of recovering the support that are more efficient. One alternative is to construct the connection graph G and then find the overlapping clustering by running a truncated power method (Yuan and Zhang, 2013) on $e_i + e_j$ (a vector that is one on indices i, j and zero elsewhere and (i, j) is an edge). In experiments, this recovers a good enough approximation to the true clustering that can then be used to smartly initialize KSVD so that it does not have to start from scratch. In practice, this yields a hybrid method that converges much more quickly and succeeds more often. Thus we feel that in practice the best algorithm may use algorithmic ideas presented here.

We note that for dictionary learning, making stochastic assumptions seems unavoidable. Interestingly, our experiments help to corroborate some of the assumptions. For instance, the condition $\mathbf{E}[X_i | X_i \neq 0] = 0$ used in our best analysis also seems necessary for KSVD; empirically we have seen its performance degrade when this is violated.

Acknowledgements We would like to thank Aditya Bhaskara, Tengyu Ma and Sushant Sachdeva for numerous helpful discussions throughout various stages of this work.

References

A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon. Learning sparsely used overcomplete dictionaries via alternating minimization. In *arxiv:1310.7991*, 2013a.

A. Agarwal, A. Anandkumar, and P. Netrapalli. Exact recovery of sparsely used overcomplete dictionaries. In *arxiv:1309.1952*, 2013b.

M. Aharon. Overcomplete dictionaries for sparse representation of signals. In *PhD Thesis*, 2006.

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. In *IEEE Trans. on Signal Processing*, pages 4311–4322, 2006.

N. Alon and D. Kleitman. Piercing convex sets and the hadwigder debrunner (p, q)-problem. In *Advances in Mathematics*, pages 103–112, 1992.

A. Anandkumar, D. Foster, D. Hsu, S. Kakade, and Y. Liu. A spectral algorithm for latent dirichlet allocation. In *NIPS*, pages 926–934, 2012.

S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – provably. In *STOC*, pages 145–162, 2012a.

S. Arora, R. Ge, and A. Moitra. Learning topic models - going beyond svd. In *FOCS*, pages 1–10, 2012b.

S. Arora, R. Ge, S. Sachdeva, and G. Schoenebeck. Finding overlapping communities in social networks: Towards a rigorous approach. In *EC*, 2012c.

M. Balcan, C. Borgs, M. Braverman, J. Chayes, and S-H Teng. Finding endogenously formed communities. In *SODA*, 2013.

Boaz Barak, John Kelner, and David Steurer. Dictionary learning using sum-of-square hierarchy. *unpublished manuscript*, 2014.

M. Belkin and K. Sinha. Polynomial learning of distribution families. In *FOCS*, pages 103–112, 2010.

E. Candes and T. Tao. Decoding by linear programming. In *IEEE Trans. on Information Theory*, pages 4203–4215, 2005.

E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. In *Communications of Pure and Applied Math*, pages 1207–1223, 2006.

P. Comon. Independent component analysis: A new concept? In *Signal Processing*, pages 287–314, 1994.

G. Davis, S. Mallat, and M. Avellaneda. Greedy adaptive approximations. In *J. of Constructive Approximation*, pages 57–98, 1997.

D. Donoho and M. Elad. Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ_1-minimization. In *PNAS*, pages 2197–2202, 2003.

D. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. In *IEEE Trans. on Information Theory*, pages 2845–2862, 1999.

D. Donoho and P. Stark. Uncertainty principles and signal recovery. In *SIAM J. on Appl. Math*, pages 906–931, 1999.

M. Elad. Sparse and redundant representations. In *Springer*, 2010.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. In *IEEE Trans. on Signal Processing*, pages 3736–3745, 2006.

K. Engan, S. Aase, and J. Hakon-Husoy. Method of optimal directions for frame design. In *ICASSP*, pages 2443–2446, 1999.

A. Frieze, M. Jerrum, and R. Kannan. Learning linear transformations. In *FOCS*, pages 359–368, 1996.

Q. Geng, H. Wang, and J. Wright. On the local correctness of ℓ_1-minimization for dictionary learning. In *arxiv:1101.5672*, 2013.

A. Gilbert, S. Muthukrishnan, and M. Strauss. Approximation of functions over redundant dictionaries using coherence. In *SODA*, 2003.

G. Golub and C. van Loan. Matrix computations. In *The Johns Hopkins University Press*, 1996.

I. J. Goodfellow, A. Courville, and Y.Bengio. Large-scale feature learning with spike-and-slab sparse coding. In *ICML*, pages 718–726, 2012.

N. Goyal, S. Vempala, and Y. Xiao. Fourier pca. In *STOC*, 2014.

R. Gribonval and M. Nielsen. Sparse representations in unions of bases. In *IEEE Transactions on Information Theory*, pages 3320–3325, 2003.

D. Gross. Recovering low-rank matrices from few coefficients in any basis. In *arxiv:0910.1879*, 2009.

D. Hanson and F. Wright. A bound on tail probabilities for quadratic forms in independent random variables. In *Annals of Math. Stat.*, pages 1079–1083, 1971.

M. Hardt. On the provable convergence of alternating minimization for matrix completion. In *arxiv:1312.0925*, 2013.

R. Horn and C. Johnson. Matrix analysis. In *Cambridge University Press*, 1990.

P. Jain, P. Netrapalli, and S. Sanghavi. Low rank matrix completion using alternating minimization. In *STOC*, pages 665–674, 2013.

K. Kavukcuoglu, M. Ranzato, and Y. LeCun. Fast inference in sparse coding algorithms with applications to object recognition. In *NYU Tech Report*, 2008.

K. Kreutz-Delgado, J. Murray, K. Engan B. Rao, T. Lee, and T. Sejnowski. Dictionary learning algorithms for sparse representation. In *Neural Computation*, 2003.

L. De Lathauwer, J Castaing, and J. Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. In *IEEE Trans. on Signal Processing*, pages 2965–2973, 2007.

H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

M. Lewicki and T. Sejnowski. Learning overcomplete representations. In *Neural Computation*, pages 337–365, 2000.

J. Mairal, M. Leordeanu, F. Bach, M. Herbert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *ECCV*, 2008.

S. Mallat. A wavelet tour of signal processing. In *Academic-Press*, 1998.

J. Matousek. Lectures on discrete geometry. In *Springer*, 2002.

A. Moitra and G. Valiant. Setting the polynomial learnability of mixtures of gaussians. In *FOCS*, pages 93–102, 2010.

B. Olshausen and B. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? In *Vision Research*, pages 3331–3325, 1997.

M. Pontil, A. Argyriou, and T. Evgeniou. Multi-task feature learning. In *NIPS*, 2007.

M. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *NIPS*, 2007.

M. Rudelson. Random vectors in the isotropic position. In *J. of Functional Analysis*, pages 60–72, 1999.

D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *Journal of Machine Learning Research*, 2012.

J. Tropp. Greed is good: Algorithmic results for sparse approximation. In *IEEE Transactions on Information Theory*, pages 2231–2242, 2004.

J. Tropp, A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *IEEE International Conf. on Image Processing*, 2003.

P. Wedin. Perturbation bounds in connection with singular value decompositions. In *BIT*, pages 99–111, 1972.

J. Yang, J. Wright, T. Huong, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008.

X. Yuan and T. Zhang. Truncated power method for sparse eigenvalue problems. In *Journal of Machine Learning Research*, pages 899–925, 2013.

Appendix A. Concentration and Clustering

Here we give the deferred proofs from Section 2.1 and Section 2.2. Before proving Lemma 7, first we state Hanson-Wright inequality:

Theorem 24 (Hanson-Wright) *(Hanson and Wright, 1971) Let X be a vector of independent, sub-Gaussian random variables with mean zero and variance one. Let M be a symmetric matrix. Then*

$$Pr[|X^T MX - tr(M)| > t] \leq 2exp\{-c\min(t^2/\|M\|_F^2, t/\|M\|_2)\}$$

Now we prove Lemma 7:

Proof: Let N be the $k \times k$ submatrix resulting from restricting $A^T A$ to the locations where $X^{(i)}$ and $X^{(j)}$ are non-zero. Set M to be a $2k \times 2k$ matrix where the $k \times k$ submatrices in the top-left and bottom-right are zero, and the $k \times k$ submatrices in the bottom-left and top-right are $(1/2)N$ and $(1/2)N^T$ respectively. Here we think of the vector X as being a length $2k$ vector whose first k entries are the non-zero entries in $X^{(i)}$ and whose last k entries are the non-zero entries in $X^{(j)}$. And by construction, we have that

$$\langle Y^{(i)}, Y^{(j)}\rangle = X^T MX$$

We can now appeal to the Hanson-Wright inequality (above). Note that since $\Omega^{(i)}$ and $\Omega^{(j)}$ do not intersect, the entries in M are each at most $\mu/\sqrt{n}$ and so the Frobenius norm of M is at most $\frac{\mu k}{\sqrt{2n}}$. This is also an upper-bound on the spectral norm of M. We can set $t = 1/2$, and for $k\mu < \sqrt{n}/C' \log n$ both terms in the minimum are $\Omega(\log n)$ and this implies the lemma. ■

We will also make use of a weaker bound (but whose conditions allow us to make fewer distributional assumptions):

Lemma 25 *If $k^2\mu < \sqrt{n}/2$ then $|\langle Y^{(i)}, Y^{(j)}\rangle| > 1/2$ implies that $\Omega^{(i)}$ and $\Omega^{(j)}$ intersect*

Proof: Suppose $\Omega^{(i)}$ and $\Omega^{(j)}$ are disjoint. Then the following upper bound holds:

$$|\langle Y^{(i)}, Y^{(j)}\rangle| \leq \sum_{p\neq q} |\langle A_p, A_q\rangle X_p^{(i)} X_q^{(j)}| \leq k^2\mu/\sqrt{n} < 1/2$$

and this implies the lemma. ■

This only works up to $k = O(n^{1/4}/\sqrt{\mu})$. In comparison, the stronger bound of Lemma 7 makes use of the randomness of the signs of X and works up to $k = O(\sqrt{n}/\mu \log n)$. Next we prove Claim 10:

Proof: Using ideas similar to Lemma 7, we can show if $|\Omega \cap \Omega^{(1)}| = 1$ (that is, the new sample has a *unique* intersection with $\Omega^{(1)}$), then $|\langle Y, Y^{(1)}\rangle| > 1/2$.

Now let $i \in \Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)}$, let $\mathcal{E}$ be the event that $\Omega \cap \Omega^{(1)} = \Omega \cap \Omega^{(2)} = \Omega \cap \Omega^{(3)} = \{i\}$. Clearly, when event $\mathcal{E}$ happens, for all $j = 1, 2, 3$, $|\langle Y, Y^{(j)}\rangle| > 1/2$. The probability of $\mathcal{E}$ is at least

$$\mathbf{Pr}[i \in \Omega]\,\mathbf{Pr}[(\Omega^{(1)} \cup \Omega^{(2)} \cup \Omega^{(3)}\backslash\{i\}) \cap \Omega = \emptyset | i \in \Omega] = k/m \cdot (1 - O(k/m) \cdot 3k) \geq k/2m.$$

Here we used bounded second moment property for the conditional probability and union bound. ■

Finally, we prove Lemma 11:

Proof: We can break up the event whose probability we would like to bound into two (not necessarily disjoint) events: (1) the probability that Ω intersects each of $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$ disjointly (i.e. it contains a point $i \in \Omega^{(1)}$ but $i \notin \Omega^{(2)}, \Omega^{(3)}$, and similarly for the other sets). (2) the probability that Ω contains a point in the common intersection of two of the sets, and one point from the remaining set. Clearly if Ω intersects the each of $\Omega^{(1)}$, $\Omega^{(2)}$ and $\Omega^{(3)}$ then at least one of these two events must occur.

The probability of the first event is at most the probability that Ω contains at least one element from each of three disjoint sets of size at most k. The probability that Ω contains an element of just one such set is at most the expected intersection which is $\frac{k^2}{m}$, and since the expected intersection of Ω with each of these sets are non-positively correlated (because they are disjoint) we have that the probability of the first event can be bounded by $\frac{k^6}{m^3}$.

Similarly, for the second event: consider the probability that Ω contains an element in $\Omega^{(1)} \cap \Omega^{(2)}$. Since $\Omega^{(1)} \cap \Omega^{(2)} \cap \Omega^{(3)} = \emptyset$, then Ω must also contain an element in $\Omega^{(3)}$ too. The expected intersection of Ω and $\Omega^{(1)} \cap \Omega^{(2)}$ is $\frac{ka}{m}$ and the expected intersection of Ω and $\Omega^{(3)}$ is $\frac{k^2}{m}$, and again the expectations are non-positively correlated since the two sets $\Omega^{(1)} \cap \Omega^{(2)}$ and $\Omega^{(3)}$ are disjoint by assumption. Repeating this argument for the other pairs completes the proof of the lemma. ■

A.1. Using Only Bounded 3-wise Moment

When the support of X has only bounded 3-wise moment, it is possible to have two supports Ω with large intersection. In that case checking the number of common neighbors cannot correctly identify whether the three samples have a common intersection. In particular, there might be false positives (three samples with no common intersection but has many common neighbors) but no false negatives (still all samples with common intersection will have many common neighbors). The algorithm can still work in this case, because it is unlikely for the two supports to have a very large intersection:

Lemma 26 *Suppose Γ has bounded 3-wise moments, $k = cm^{2/5}$ for some small enough constant $c > 0$. For any set Ω of size k, the probability that a random support Ω' from Γ has intersection larger than $m^{1/5}/100$ with Ω is at most $O(m^{-6/5})$.*

Proof: Let T be the number of triples in the intersection of Ω and Ω'. For any triple in Ω, the probability that it is also in Ω' is at most $O(k^3/m^3)$ by bounded 3-wise moment. Therefore $\mathbf{E}[T] \le \binom{k}{3} O(k^3/m^3) = O(k^6/m^3)$.

On the other hand, whenever Ω and Ω' has more than $m^{1/5}/100$ intersections, T is larger than $\binom{m^{1/5}/100}{3}$. By Markov's inequality we know $\mathbf{Pr}[|\Omega \cap \Omega'| \ge m^{1/5}/100] \le O(m^{-6/5})$. ■

Since the probability of having false positives is small (but not negligible), we can do a simple trimming operation when we are computing the set $S_{u,v}$ in Algorithm 1. We shall change the definition of $S_{u,v}$ as follows:

1. Set $S'_{u,v} = \{w : |\Gamma_G(u) \cap \Gamma_G(v) \cap \Gamma_G(w)| \ge T\} \cup \{u, v\}$.

2. Set $S_{u,v} = \{w : w \in S'_{u,v} \text{ and } |\Gamma_G(w) \cap S'_{u,v}| \ge T\}$.

Now $S'_{u,v}$ is the same as the old definition and may have false positives. However, intuitively the false positives are not in the cluster so they cannot have many connections to the cluster, and will be filtered out in the second step. In particular, we have the following lemma:

Lemma 27 *If (u, v) is an indentifying pair (as defined in Definition 12) for i, then with high probability $S_{u,v}$ is the set $\mathcal{C}_i = \{j : i \in \Omega^{(j)}\}$.*

Proof: First we argue the set $S'_{u,v}$ is the union of $\mathcal{C}_i$ with a small set. By Claim 10 and Chernoff bound, for all $w \in \mathcal{C}_i$ u, v, w has more than T common neighbors, so $w \in S'_{u,v}$. On the other hand, if $w \notin \mathcal{C}_i$ but $w \in S'_{u,v}$, then by Lemma 11 we know $\Omega^{(w)}$ must have a large intersection with either $\Omega^{(u)}$ or $\Omega^{(v)}$, which has probability only $O(m^{-6/5})$ by Lemma 26. Therefore again by concentration bounds with high probability $|S'_{u,v} \backslash \mathcal{C}_i| \le p/m \ll T$.

Now consider the second step. For the samples in $\mathcal{C}_i$, the probability that they are connected to another random sample in $\mathcal{C}_i$ is $1-O(k^2/m)$, so by concentration bounds with high probability they have at least T neighbors in $\mathcal{C}_i$, and they will not be filtered and are still in $S_{u,v}$. On the other hand, for any vertex $w \notin \mathcal{C}_i$, the expected number of edges from w to $\mathcal{C}_i$ is only $O(k^2/m)|\mathcal{C}_i| \ll T$, and by concentration property, they are concentrated around the expectation with high probability. So for any $w \in S'_{u,v} \backslash \mathcal{C}_i$, it can only have $O(pk^3/m^2)$ edges to $\mathcal{C}_i$, and $O(p/m)$ edges to $S'_{u,v} \backslash \mathcal{C}_i$. The total number of edges to $S'_{u,v}$ is much less than T, so all of those vertices are going to be removed, and $S_{u,v} = \mathcal{C}$. ■

This lemma ensures after we pick enough random pairs, with high probability all the correct clusters $\mathcal{C}_i$'s are among the $S_{u,v}$'s. There can be "bad" sets, but same as before all those sets contains some of the $\mathcal{C}_i$, so will be removed at the end of the algorithm:

Claim 28 *For any pair (u, v) with $i \in \Omega^{(u)} \cap \Omega^{(v)}$, let $\mathcal{C}_i = \{j : i \in \Omega^{(j)}\}$, then with high probability $\mathcal{C}_i \subseteq S_{u,v}$.*

Proof: This is essentially in the proof of the previous lemma. As before by Claim 10 we know $\mathcal{C}_i \subseteq S'_{u,v}$. Now for any sample in $\mathcal{C}_i$, the expected number of edges to $\mathcal{C}_i$ is $(1 - o(1))|\mathcal{C}_i|$, by concentration bounds we know the number of neighbors is larger than T with high probability. Then we apply union bound for all samples in $\mathcal{C}_i$, and conclude that $\mathcal{C}_i \subseteq S_{u,v}$. ■

Appendix B. Finding the Relative Signs

Here we prove Theorem 14.

Lemma 29 *In Algorithm 3, $\mathcal{C}_i^{\pm}$ is either $\{u : X_i^{(u)} > 0\}$ or $\{u : X_i^{(u)} < 0\}$.*

Proof: It suffices to prove the lemma at the start of Step 8, since this step only takes the complement of $\mathcal{C}_i^{\pm}$ with respect to $\mathcal{C}_i$. Appealing to Lemma 7 we conclude that $\Omega^{(u)}$ and $\Omega^{(v)}$ uniquely intersect in coordinate i then the sign of $\langle Y^{(u)}, Y^{(v)} \rangle$ is equal to the sign of $X_i^{(u)} X_i^{(v)}$. Hence when Algorithm 3 adds an element to $\mathcal{C}_i^{\pm}$ it must have the same sign as the i^{th} component of $X^{(u_i)}$. What remains is to prove that each node $v \in \mathcal{C}_i$ is correctly labeled. We will do this by showing that for any such vertex, there is a length two path of labeled pairs that connects u_i to v, and this is true because the number of labeled pairs is large. We need the following simple claim:

Algorithm 3 OVERLAPPINGAVERAGE, **Input:** p samples $Y^{(1)}, Y^{(2)}, ...Y^{(p)}$ and overlapping clusters $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_m$

```
For each $\mathcal{C}_i$
    For each pair $(u, v) \in \mathcal{C}_i$ that does not appear in any other $\mathcal{C}_j$ ($X^{(u)}$ and $X^{(v)}$ have a unique intersection)
        Label the pair $+1$ if $\langle Y^{(u)}, Y^{(v)} \rangle > 0$ and otherwise label it $-1$.
    Choose an arbitrary $u_i \in \mathcal{C}_i$, and set $\mathcal{C}_i^{\pm} = \{u_i\}$
    For each $v \in \mathcal{C}_i$
        If the pair $u_i, v$ is labeled $+1$ add $v$ to $\mathcal{C}_i^{\pm}$
        Else if there is $w \in \mathcal{C}_i$ where the pairs $u_i, w$ and $v, w$ have the same label, add $v$ to $\mathcal{C}_i^{\pm}$.
    If $|\mathcal{C}_i^{\pm}| \leq |\mathcal{C}_i|/2$ set $\mathcal{C}_i^{\pm} \leftarrow \mathcal{C}_i \backslash \mathcal{C}_i^{\pm}$.
    Let $\hat{A}_i = \sum_{v \in \mathcal{C}_i^{\pm}} Y^{(v)} / \| \sum_{v \in \mathcal{C}_i^{\pm}} Y^{(v)} \|$
Output $\hat{A}$, where each column is $\hat{A}_i$ for some $i$
```

Claim 30 *If $p > m^2 \log m/k^2$ then with high probability any two clusters share at most $2pk^2/m^2$ nodes in common.*

This follows since the probability that a node is contained in any fixed pair of clusters is at most k^2/m^2. Then for any node $u \in \mathcal{C}_i$, we would like to lower bound the number of labeled pairs it has in $\mathcal{C}_i$. Since u is in at most $k-1$ other clusters $\mathcal{C}_{i_1}, ..., \mathcal{C}_{i_{k-1}}$, the number of pairs u, v where $v \in \mathcal{C}_i$ that are not labeled for $\mathcal{C}_i$ is at most

$$\sum_{t=1}^{k-1} |\mathcal{C}_{i_t} \cap \mathcal{C}_i| \leq k \cdot 2pk^2/m^2 \ll pk/3m = |\mathcal{C}_i|/3$$

Therefore for a fixed node u for at least a $2/3$ fraction of the other nodes $w \in \mathcal{C}_i$ the pair u, w is labeled. Hence we conclude that for each pair of nodes $u_i, v \in \mathcal{C}_i$ the number of w for which both u_i, w and w, v are labeled is at least $|\mathcal{C}_i|/3 > 0$ and so for every v, there is a labeled path of length two connecting u_i to v. ■

Using this lemma, we are ready to prove Algorithm 3 correctly learns all columns of A.

Proof: We can invoke Lemma 29 and conclude that $\mathcal{C}_i^{\pm}$ is either $\{u : X_i^{(u)} > 0\}$ or $\{u : X_i^{(u)} < 0\}$, whichever set is larger. Let us suppose that it is the former. Then each $Y^{(u)}$ in $\mathcal{C}_i^{\pm}$ is an independent sample from the distribution conditioned on $X_i > 0$, which we call Γ_i^+. We have that $\mathbf{E}_{\Gamma_i^+}[AX] = cA_i$ where c is a constant in $[1, C]$ because $\mathbf{E}_{\Gamma_i^+}[X_j] = 0$ for all $j \neq i$.

Let us compute the variance:

$$\mathop{\mathbf{E}}_{\Gamma_i^+}[\|AX - \mathop{\mathbf{E}}_{\Gamma_i^+} AX\|^2] \leq \mathop{\mathbf{E}}_{\Gamma_i^+} X_i^2 + \sum_{j \neq i} \mathop{\mathbf{E}}_{\Gamma_i^+}[X_j^2] \leq C^2 + \sum_{j \neq i} C^2 k/m \leq C^2(k+1),$$

Note that there are no cross-terms because the signs of each X_j are independent. Furthermore we can bound the norm of each vector $Y^{(u)}$ via incoherence. We can conclude that if $|\mathcal{C}_i^{\pm}| >$

$C^2 k \log m/\epsilon^2$, then with high probability $\|\hat{A}_i - A_i\| \le \epsilon$ using vector Bernstein's inequality (Gross (2009), Theorem 12). This latter condition holds because we set $\mathcal{C}_i^{\pm}$ to itself or its complement based on which one is larger. ■

Appendix C. An Approach via SVD

Here we give an alternative algorithm for recovering the dictionary based instead on SVD. Let us fix some notation: Let Γ_i be the distribution conditioned on $X_i \neq 0$. Then once we have found the overlapping clustering, each cluster is a set of random samples from Γ_i. Also let $\alpha = |\langle u, A_i\rangle|$.

Definition 31 *Let* $R_i^2 = 1 + \sum_{j\neq i} \langle A_i, A_j\rangle^2 E_{\Gamma_i}[X_j^2]$.

Note that R_i^2 is the projected variance of Γ_i on the direction $u = A_i$. Our goal is to show that for any $u \neq A_i$ (i.e. $\alpha \neq 1$), the variance is strictly smaller.

Lemma 32 *The projected variance of* Γ_i *on* u *is at most*

$$\alpha^2 R_i^2 + \alpha\sqrt{(1-\alpha^2)}\frac{2\mu k}{\sqrt{n}} + (1-\alpha^2)(\frac{k}{m} + \frac{\mu k}{\sqrt{n}})$$

Proof: Let $u^{||}$ and $u^{\perp}$ be the components of u in the direction of A_i and perpendicular to A_i. Then we want bound $E_{\Gamma_i}[\langle u, Y\rangle^2]$ where Y is sampled from Γ_i. Since the signs of each X_j are independent, we can write

$$E_{\Gamma_i}[\langle u, Y\rangle^2] = \sum_j E_{\Gamma_i}[\langle u, A_j X_j\rangle^2] = \sum_j E_{\Gamma_i}[\langle u^{||} + u^{\perp}, A_j X_j\rangle^2]$$

Since $\alpha = \|u^{||}\|$ we have:

$$E_{\Gamma_i}[\langle u, Y\rangle^2] = \alpha^2 R_i^2 + E_{\Gamma_i}[\sum_{j\neq i}(2\langle u^{||}, A_j\rangle\langle u^{\perp}, A_j\rangle + \langle u^{\perp}, A_j\rangle^2)X_j^2]$$

Also $E_{\Gamma_i}[X_j^2] = (k-1)/(m-1)$. Let v be the unit vector in the direction $u^{\perp}$. We can write

$$E_{\Gamma_i}[\sum_{j\neq i}\langle u^{\perp}, A_j\rangle^2 X_j^2] = (1-\alpha^2)(\frac{k-1}{m-1})v^T A_{-i} A_{-i}^T v$$

where A_{-i} denotes the dictionary A with the i^{th} column removed. The maximum over v of $v^T A_{-i} A_{-i}^T v$ is just the largest singular value of $A_{-i} A_{-i}^T$ which is the same as the largest singular value of $A_{-i}^T A_{-i}$ which by the Greshgorin Disk Theorem (see e.g. Horn and Johnson (1990)) is at most $1 + \frac{\mu}{\sqrt{n}} m$. And hence we can bound

$$E_{\Gamma_i}[\sum_{j\neq i}\langle u^{\perp}, A_j\rangle^2 X_j^2] \le (1-\alpha^2)(\frac{k}{m} + \frac{\mu k}{\sqrt{n}})$$

Also since $|\langle u^{||}, A_j\rangle| = \alpha|\langle A_i, A_j\rangle| \le \alpha\mu/\sqrt{n}$ we obtain:

$$E[\sum_{j\neq i} 2\langle u^{||}, A_j\rangle\langle u^{\perp}, A_j\rangle X_j^2] \le \alpha\sqrt{(1-\alpha^2)}\frac{2\mu k}{\sqrt{n}}$$

and this concludes the proof of the lemma. ■

Algorithm 4 OVERLAPPINGSVD, **Input:** p samples $Y^{(1)}, Y^{(2)}, ...Y^{(p)}$

```
1. Run OVERLAPPINGCLUSTER (or OVERLAPPINGCLUSTER2) on the $p$ samples
2. Let $\mathcal{C}_1, \mathcal{C}_2, ...\mathcal{C}_m$ be the $m$ returned overlapping clusters
3. Compute $\hat{\Sigma}_i = \frac{1}{|\mathcal{C}_i|}\sum_{Y\in\mathcal{C}_i} YY^T$
4. Compute the first singular value $\hat{A}_i$ of $\hat{\Sigma}_i$
5. Output $\hat{A}$, where each column is $\hat{A}_i$ for some $i$
```

Definition 33 *Let $\zeta = \max\{\frac{\mu k}{\sqrt{n}}, \sqrt{\frac{k}{m}}\}$, so the expression in Lemma 32 can be be an upper bounded by $\alpha^2 R_i^2 + 2\alpha\sqrt{1-\alpha^2}\cdot\zeta + (1-\alpha^2)\zeta^2$.*

We will show that an approach based on SVD recovers the true dictionary up to additive accuracy $\pm\zeta$. Note that here ζ is a parameter that converges to zero as the size of the problem increases, but is not a function of the number of samples. So unlike the algorithm in the previous subsection, we cannot make the error in our algorithm arbitrarily small by increasing the number of samples, but this algorithm has the advantage that it succeeds even when $\mathbf{E}[X_i] \neq 0$.

Corollary 34 *The maximum singular value of Γ_i is at least R_i and the direction u satisfies $\|u - A_i\| \le O(\zeta)$. Furthermore the second largest singular value is bounded by $O(R_i^2\zeta^2)$.*

Proof: The bound in Lemma 32 is only an upper bound, however the direction $\alpha = 1$ has variance $R_i^2 > 1$ and hence the direction of maximum variance must correspond to $\alpha \in [1-O(\zeta^2), 1]$. Then we can appeal to the variational characterization of singular values (see Horn and Johnson (1990)) that

$$\sigma_2(\Sigma_i) = \max_{u\perp A_i} \frac{u^T\Sigma_i u}{u^T u}$$

Then condition that $\alpha \in [-O(\zeta), O(\zeta)]$ for the second singular value implies the second part of the corollary. ■

Since we have a lower bound on the separation between the first and second singular values of Σ_i, we can apply Wedin's Theorem and show that we can recover A_i approximately even in the presence of noise.

Theorem 35 (Wedin) *(Wedin, 1972) Let $\delta = \sigma_1(M) - \sigma_2(M)$ and let $M' = M + E$ and furthermore let v_1 and v_1' be the first singular vectors of M and M' respectively. Then*

$$\sin\Theta(v_1, v_1') \le C\frac{\|E\|}{\delta}$$

Hence even if we do not have access to Σ_i but rather an approximation to it $\hat{\Sigma}_i$ (e.g. an empirical covariance matrix computed from our samples), we can use the above perturbation bound to show that we can still recover a direction that is close to A_i – and in fact converges to A_i as we take more and more samples.

Theorem 36 *If the input to* OVERLAPPINGSVD *is the correct clustering, then the algorithm outputs a dictionary $\hat{A}$ so that for each i, $\|A_i - \hat{A}_i\| \le \zeta$ with high probability if $k \le c \min(\sqrt{m}, \frac{\sqrt{n}}{\mu \log n})$ and if*

$$p \ge \max(m^2 \log m/k^2, \frac{mn \log m \log n}{\zeta^2})$$

Proof: Appealing to Theorem 13, we have that with high probability the call to OVERLAPPINGCLUSTER returns the correct overlapping clustering. Then given $\frac{n \log n}{\zeta^2}$ samples from the distribution Γ_i the classic result of Rudelson implies that the computed empirical covariance matrix $\hat{\Sigma}_i$ is close in spectral norm to the true co-variance matrix Rudelson (1999). This, combined with the separation of the first and second singular values established in Corollary 34 and Wedin's Theorem 35 imply that we recover each column of A up to an additive accuracy of ϵ and this implies the theorem. Note that since we only need to compute the first singular vector, this can be done via power iteration Golub and van Loan (1996) and hence the bottleneck in the running time is the call to OVERLAPPINGCLUSTER. ■

Appendix D. Refining the Solution

In this section, we prove the deferred claims from Section 4. First we prove Claim 16:

Proof: We can compute $\langle Y^{(i)}, B_1\rangle = \sum_{j\in\Omega^{(i)}} X_j^{(i)} \langle A_j, B_1\rangle$ and the total contribution of all of the terms besides $X_1^{(i)}\langle A_1, B_1\rangle$ for $j \neq 1$ is at most $1/3$. This implies the claim. ■

Then we prove Lemma 20:

Proof: Let us first compute $\hat{B}_1 - B_1$:

$$\begin{aligned}\hat{B}_1 - B_1 &= \frac{\sum_{i=1}^{l} X_1^{(i)}((I - M_i)A_1 - B_1) + \sum_{i=1}^{l} \sum_{j\in\Omega^{(i)}\setminus\{1\}} (I - M_i)A_j X_j^{(i)}}{\sum_{i=1}^{l} X_1^{(i)}} \\ &= \sum_{i=1}^{l} \frac{X_1^{(i)}}{\sum_{i=1}^{l} X_1^{(i)}} (I - M_i)(A_1 - B_1) + \frac{\sum_{i=1}^{l} \sum_{j\in\Omega^{(i)}\setminus\{1\}} (I - M_i)(A_j - B_j)X_j^{(i)}}{\sum_{i=1}^{l} X_1^{(i)}}.\end{aligned}$$

The last equality uses the first and second properties of M_i from the above claim. Consequently we have

$$\begin{aligned}A_1 - \hat{B}_1 &= (A_1 - B_1) - (\hat{B}_1 - B_1) \\ &= \sum_{i=1}^{l} \frac{X_1^{(i)}}{\sum_{i=1}^{l} X_1^{(i)}} M_i(A_1 - B_1) - \frac{\sum_{i=1}^{l} \sum_{j\in\Omega^{(i)}\setminus\{1\}} (I - M_i)(A_j - B_j)X_j^{(i)}}{\sum_{i=1}^{l} X_1^{(i)}}.\end{aligned}$$

And this is our desired expression. ■

Next we prove Claim 21:

Proof: The denominator is at least l and the numeratoris the sum of at most lk independent random vectors with mean zero, and whose length is at most $3C\epsilon$. We can invoke the vector Bernstein's

Algorithm 5 OVERLAPPINGCLUSTER2, **Input:** p samples $Y^{(1)}, Y^{(2)}, ..., Y^{(p)}$, integer ℓ

```
Compute a graph $G$ on $p$ nodes where there is an edge between $i$ and $j$ iff $|\langle Y^{(i)}, Y^{(j)} \rangle| > 1/2$
Set $T = \frac{pk}{Cm2^\ell}$
Repeat $\Omega(k^{\ell-2} m \log^2 m)$ times:
    Choose a random node $u$ in $G$, and $\ell - 1$ neighbors $u_1, u_2, ... u_{\ell-1}$
    If $|\Gamma_G(u) \cap \Gamma_G(u_1) \cap ... \cap \Gamma_G(u_{\ell-1})| \geq T$
        Set $S_{u_1,u_2,...u_{\ell-1}} = \{w \text{ s.t. } |\Gamma_G(u) \cap \Gamma_G(u_1) \cap ... \cap \Gamma_G(w)| \geq T\} \cup \{u_1, u_2, ... u_{\ell-1}\}$
Delete any set $S_{u_1,u_2,...u_{\ell-1}}$ if $u_1, u_2, ... u_{\ell-1}$ are contained in a strictly smaller set $S_{v_1,v_2,...v_{\ell-1}}$
Output the remaining sets $S_{u_1,u_2,...u_{\ell-1}}$
```

inequality Gross (2009), and conclude that the sum is bounded by $O(C\sqrt{lk} \log m\epsilon)$ with high probability. After normalization the second term is bounded by $\epsilon/100$. ■

Finally, we prove Claim 23:

Proof: We have that $\|x - y\|^2 = \sin^2 \theta + (1 - \cos \theta)^2$ where θ is the angle between x and y. Note that $\sin \theta \leq \|x' - y\| \leq \epsilon$ so hence $\|x - y\| \leq \sqrt{\epsilon^2 + (1 - \sqrt{1 - \epsilon^2})^2}$. Note that for $0 \leq a \leq 1$ we have $1 - a \leq \sqrt{1 - a}$ and this implies the claim. ■

Appendix E. A Higher Order Algorithm

Here we extend the algorithm OVERLAPPINGCLUSTER presented in Section 2.2 to succeed even when $k \leq c \min(m^{1/2-\eta}, \sqrt{n}/\mu \log n)$. The premise of OVERLAPPINGCLUSTER is that we can distinguish whether or not a triple of samples $Y^{(1)}, Y^{(2)}, Y^{(3)}$ has a common intersection based on their number of common neighbors in the connection graph. However for $k = \omega(m^{2/5})$ this is no longer true! But we will instead consider higher-order groups of sets. In particular, for any $\eta > 0$ there is an ℓ so that we can distinguish whether an ℓ-tuple of samples $Y^{(1)}, Y^{(2)}, ..., Y^{(\ell)}$ has a common intersection or not based on their number of common neighbors, and this test succeeds even for $k = \Omega(m^{1/2-\eta})$.

The main technical challenge is in showing that if the sets $\Omega^{(1)}, \Omega^{(2)}, ..., \Omega^{(\ell)}$ do not have a common intersection, that we can upper bound the probability that a random set Ω intersects each of them. To accomplish this, we will need to bound the number of ways of piercing ℓ sets $\Omega^{(1)}, \Omega^{(2)}, ..., \Omega^{(\ell)}$ that have bounded pairwise intersections by at most s points (see definitions below and Lemma 40), and this is the key to analyzing our higher order algorithm OVERLAPPINGCLUSTER2. We will defer the proofs of the key lemmas and the description of the algorithm in this section to Appendix E.

Nevertheless what we need is an analogue of Claim 10 and Lemma 11. The first is easy, but what about an analogue of Lemma 11? To analyze the probability that a set Ω intersects each of the sets $\Omega^{(1)}, \Omega^{(2)}, ..., \Omega^{(\ell)}$ we will rely on the following standard definition:

Definition 37 *Given a collection of sets $\Omega^{(1)}, \Omega^{(2)}, ..., \Omega^{(\ell)}$, the* piercing number *is the minimum number of points $p_1, p_2, ..., p_r$ so that each set contains at least one point p_i.*

The notion of piercing number is well-studied in combinatorics (see e.g. Matousek (2002)). However, one is usually interested in upper-bounding the piercing number. For example, a classic result of Alon and Kleitman concerns the (p, q)-problem (Alon and Kleitman, 1992): Suppose we are given a collection of sets that has the property that each choice of p of them has a subset of q which intersect. Then how large can the piercing number be? Alon and Kleitman proved that the piercing number is at most a fixed constant $c(p, q)$ independent of the number of sets (Alon and Kleitman, 1992).

However, here our interest in piercing number is not in bounding the minimum number of points needed but rather in analyzing how many ways there are of piercing a collection of sets with at most s points, since this will directly yield bounds on the probability that Ω intersects each of $\Omega^{(1)}, \Omega^{(2)}, ..., \Omega^{(\ell)}$. We will need as a condition that each pair of sets has bounded intersection, and this holds in our model with high-probability.

Claim 38 *With high probability, the intersection of any pair $\Omega^{(1)}, \Omega^{(2)}$ has size at most Q*

Definition 39 *We will call a set of ℓ sets a (k, Q) family if each set has size at most k and the intersection of each pair of sets has size at most Q.*

Lemma 40 *The number of ways of piercing (k, Q) family (of ℓ sets) with s points is at most $(\ell k)^s$. And crucially if $\ell \geq s + 1$, then the number of ways of piercing it with s points is at most $Qs(s + 1)(\ell k)^{s-1}$.*

Proof: The first part of the lemma is the obvious upper bound. Now let us assume $\ell \geq s + 1$: Then given a set of s points that pierce the sets, we can partition the ℓ sets into s sets based on which of the s points is hits the set. (In general, a set may be hit by more than one point, but we can break ties arbitrarily). Let us fix any $s+1$ of the ℓ sets, and let U be the the union of the pairwise intersections of each of these sets. Then U has size at most $Qs(s+1)$. Furthermore by the Pigeon Hole Principle, there must be a pair of these sets that is hit by the same point. Hence one of the s points must belong to the set U, and we can remove this point and appeal to the first part of the lemma (removing any sets that are hit by this point). This concludes the proof of the second part of the lemma, too. ■

Theorem 41 *The algorithm* OVERLAPPINGCLUSTER2(ℓ) *finds an overlapping clustering where each set corresponds to some i and contains all $Y^{(j)}$ for which $X_i^{(j)} \neq 0$. The algorithm runs in time $\widetilde{O}(k^{\ell-2}mp + p^2 n)$ and succeeds with high probability if $k \leq c \min(m^{(\ell-1)/(2\ell-1)}, \frac{\sqrt{n}}{\mu \log n})$ and if $p = \Omega(m^2/k^2 \log m + k^{\ell-2} m \log^2 m)$*

In order to prove this theorem we first give an analogue of Claim 10:

Claim 42 *Suppose $\Omega^{(1)} \cap \Omega^{(2)} \cap ... \cap \Omega^{(\ell)} \neq \emptyset$, then*

$$Pr_Y[\textit{for all } j = 1, 2, ..., \ell, |\langle Y, Y^{(j)} \rangle| > 1/2] \geq \frac{k}{2m}$$

The proof of this claim is identical to the proof of Claim 10. Next we give the crucial corollary of Lemma 40.

Corollary 43 *The probability that Ω hits each set in a (k, Q) family (of ℓ sets) is at most*

$$\sum_{2\leq s\leq \ell-1} (Qs(s+1)(\ell k)^{s-1})\left(\frac{k}{m}\right)^s + \sum_{s\geq \ell}\left(\frac{\ell k^2}{m}\right)^s$$

where C_s is a constant depending polynomially on s.

Proof: We can break up the probability of the event that Ω hits each set in a (k, Q) family into another family of events. Let us consider the probability that X pierces the family with $s \leq \ell - 1$ points or $s \geq \ell$ points. In the former case, we can invoke the second part of Lemma 40 and the probability that X hits any particular set of s points is at most $(k/m)^s$. In the latter case, we can invoke the first part of Lemma 40. ■

Note that if $k \leq m^{1/2}$ then k/m is always greater than or equal to $k^{s-1}(k/m)^s$. And so asymptotically the largest term in the above sum is $(k^2/m)^\ell$ which we want to be asymptotically smaller than k/m which is the probability in Claim 42. So if $k \leq cm^{(\ell-1)/(2\ell-1)}$ then above bound is $o(k/m)$ which is asymptotically smaller than the probability that a given set of ℓ nodes that have a common intersection are each connected to a random (new) node in the connection graph. So again, we can distinguish between whether or not an ℓ-tuple has a common intersection or not and this immediately yields a new overlapping clustering algorithm that works for k almost as large as $\sqrt{m}$, although the running time depends on how close k is to this bound.

Appendix F. Extensions: Proof Sketch of Theorem 6

Let us first examine how the conditions in the hypothesis of Theorem 4 were used in its proof and then discuss why they can be relaxed.

Our algorithm is based on three steps: constructing the connection graph, finding the overlapping clustering, and recovering the dictionary. However if we invoke Lemma 25 (as opposed to Lemma 7) then the properties we need of the connection graph follow from each X being at most k sparse for $k \leq n^{1/4}/\sqrt{\mu}$ without any distributional assumptions.

Furthermore, the crucial steps in finding the overlapping clustering are bounds on the probability that a sample X intersects a triple with a common intersection, and the probability that it does so when there is no common intersection (Claim 10 and Lemma 11). Indeed, these bounds hold whenever the probability of two sets intersecting in two or more locations is smaller (by, say, a factor of k) than the probability of the sets intersecting once. This can be true even if elements in the sets have significant positive correlation (but for the ease of exposition, we have emphasized the simplest models at the expense of generality). Lastly, Algorithm 3 we can instead consider the difference between the averages for S_i and $\mathcal{C}_i \backslash S_i$ and this succeeds even if $\mathbf{E}[X_i]$ is non-zero. This last step does use the condition that the variables X_i are independent, but if we instead use Algorithm 4 we can circumvent this assumption and still recover a dictionary that is close to the true one.

Finally, the "bounded away from zero" assumption in Definition 2 can be relaxed: the resulting algorithm recovers a dictionary that is close enough to the true one and still allows sparse recovery. This is because when the distribution has the anti-concentration property, a slight variant of Algorithm 1 can still find most (instead of all) columns with $X_i \neq 0$.

Using the ideas from this part, we give a proof sketch for Theorem 6

Proof:[sketch for Theorem 6] The proof follows the same steps as the proof of Theorem 36. There are a few steps that needs to be modified:

1. Invoke Lemma 25 instead of Lemma 7.
2. For Lemma 11, use the weaker bound on the 4-th moment. This is still OK because k is smaller now.
3. In Definition 31, redefine R_i^2 to be $\mathbf{E}_{x \in D^i}[\langle A_i, Ax \rangle^2]$.
4. In Lemma 32, use the bound $R_i^2 \alpha^2 + \alpha\sqrt{1-\alpha^2} 2k\sqrt{\mu}/n^{1/4} + (1-\alpha^2)k^2\mu/\sqrt{n}$ in order to take the correlations between X_i's into account.

■

Remark: Based on different assumptions on the distribution, there are algorithms with different trade-offs. Theorem 6 is only used to illustrate the potential of our approach and does not try to achieve optimal trade-off in every case.

A major difference from class Γ is that the X_i's do not have expectation 0 and are not forbidden from taking values close to 0 (provided they do have reasonable probability of taking values away from 0). Another major difference is that the distribution of X_i *can* depend upon the values of other nonzero coordinates. The weaker moment condition allows a fair bit of correlation among the set of nonzero coordinates.

It is also possible to relax the condition that each nonzero X_i is in $[-C, -1] \cup [1, C]$. Instead we require X_i has magnitude at most $O(1)$, and has a weak *anti-concentration* property: for every $\delta > 0$ it has probability at least $c_\delta > 0$ of exceeding δ in magnitude. This requires changing Algorithm 1 in the following ways:

For each set S, let T be the subset of vertices that have at least $1 - 2\delta$ neighbors in S: $T = \{i \in S, |\Gamma_G(i) \cap S| \geq (1-2\delta)|S|$. Keep sets S that $1 - 2\delta$ fraction of the vertices are in T ($|T| \geq (1-2\delta)|S|$).Here the choice of δ depend on parameters μ, n, k, and effects the final accuracy of the algorithm. This ensures for any remaining S, there must be a single coordinate that every $X^{(i)}$ for $i \in S$ is nonzero on.

In the last step, only output sets that are significantly different from the previously outputted sets (significantly different means the symmetric difference is at least $pk/5m$)

Appendix G. Discussion: Overlapping Communities

There is a connection between the approach used here, and the recent work on algorithms for finding overlapping communities (see in particular Arora et al. (2012c), Balcan et al. (2013)). We can think of the set of samples Y for which $X_i \neq 0$ as a "community". Then each sample is in more than one community, and indeed for our setting of parameters each sample is contained in k communities. We can think of the main approach of this paper as:

If we can find all of the overlapping communities, then we can learn an unknown dictionary.

So how can we find these overlapping communities? The recent papers Arora et al. (2012c), Balcan et al. (2013) pose deterministic conditions on what constitutes a community (e.g. each node outside of the community has fewer edges into the community than do other members of the community). These papers provide algorithms for finding all of the communities, provided these conditions are met. However for our setting of parameters, both of these algorithms would run in quasi-polynomial time. For example, the parameter "d" in the paper Arora et al. (2012c) is an upper-bound on how many communities a node can belong to, and the running time of the algorithms in Arora et al. (2012c) are quasi-polynomial in this parameter. But in our setting, each sample Y belongs to k communities – one for each non-zero value in X – and the most interesting setting here is when k is polynomially large. Similarly, the parameter "θ" in Balcan et al. (2013) can be thought of as: If node u is in community c, what is the ratio of the edges incident to u that leave the community c compared to the number that stay inside c? Again, for our purposes this parameter "θ" is roughly k and the algorithms in Balcan et al. (2013) depend quasi-polynomially on this parameter.

Hence these algorithms would not suffice for our purposes because when applied to learning an unknown dictionary, their running time would depend quasi-polynomially on the sparsity k. In contrast, our algorithms run in polynomial time in all of the parameters, albeit for a more restricted notion of what constitutes a community (but one that seems quite natural from the perspective of dictionary learning). Our algorithm OVERLAPPINGCLUSTER finds all of the overlapping "communities" provided that whenever a triple of nodes shares a common community they have many more common neighbors than if they do not all share a single community. The correctness of the algorithm is quite easy to prove, once this condition is met; but here the main work was in showing that our generative model meets these neighborhood conditions.