# Exact Recovery of Sparsely-Used Dictionaries

**Daniel A. Spielman** SPIELMAN@CS.YALE.EDU
**Huan Wang** HUAN.WANG@YALE.EDU
*Department of Computer Science, Yale University*

**John Wright** JOHNWRIGHT@EE.COLUMBIA.EDU
*Department of Electrical Engineering, Columbia University*

**Editor:** Shie Mannor, Nathan Srebro, Robert C. Williamson

## Abstract

We consider the problem of learning sparsely used dictionaries with an arbitrary square dictionary and a random, sparse coefficient matrix. We prove that $O(n \log n)$ samples are sufficient to uniquely determine the coefficient matrix. Based on this proof, we design a polynomial-time algorithm, called Exact Recovery of Sparsely-Used Dictionaries (ER-SpUD), and prove that it probably recovers the dictionary and coefficient matrix when the coefficient matrix is sufficiently sparse. Simulation results show that ER-SpUD reveals the true dictionary as well as the coefficients with probability higher than many state-of-the-art algorithms.

**Keywords:** Dictionary learning, matrix decomposition, matrix sparsification.

## 1. Introduction

In the Sparsely-Used Dictionary Learning Problem, one is given a matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times p}$ and asked to find a pair of matrices $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{X} \in \mathbb{R}^{m \times p}$ so that $\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|$ is small and so that $\boldsymbol{X}$ is *sparse* – $\boldsymbol{X}$ has only a few nonzero elements. We examine solutions to this problem in which $\boldsymbol{A}$ is a basis, so $m = n$, and without the presence of noise, in which case we insist $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$. Variants of this problem arise in different contexts in machine learning, signal processing, and even computational neuroscience. We list two prominent examples:

- *Dictionary learning [16; 12]:* Here, the goal is to find a basis $\boldsymbol{A}$ that most compactly represents a given set of sample data. Techniques based on learned dictionaries have performed quite well in a number of applications in signal and image processing [3; 18; 20].
- *Blind source separation [22]:* Here, the rows of $\boldsymbol{X}$ are considered the emissions of various sources over time. The sources are linearly mixed by $\boldsymbol{A}$ (instantaneous mixing). Sparse component analysis [22; 8] is the problem of using the prior information that the sources are sparse in some domain to unmix $\boldsymbol{Y}$ and obtain $(\boldsymbol{A}, \boldsymbol{X})$.

These applications raise several basic questions. First, when is the problem well-posed? More precisely, suppose that $\boldsymbol{Y}$ is indeed the product of some unknown dictionary $\boldsymbol{A}$ and sparse coefficient matrix $\boldsymbol{X}$. Is it possible to identify $\boldsymbol{A}$ and $\boldsymbol{X}$, up to scaling and permutation. If we assume that the rows of $\boldsymbol{X}$ are sampled from independent random sources, classical, general results in the literature on Independent Component Analysis imply that the problem is solvable in the large sample limit [4]. If we instead assume that the columns of $\boldsymbol{X}$ each have at most $k$ nonzero entries, and that for each possible pattern of nonzeros, we have observed $k+1$ nondegenerate samples $\boldsymbol{y}_j$, the problem is again well-posed [13; 8]. This suggests a sample requirement of $p \geq (k+1)\binom{n}{k}$. We ask: is this large number necessary? Or could it be that the desired factorization is unique[1] even with more realistic sample sizes?

Second, suppose that we know that the problem is well-posed. Can it be solved efficiently? This question has been vigorously investigated by many authors, starting from seminal work of Olshausen and Field [16], and continuing with the development of alternating directions methods such as the Method of Optimal Directions (MOD) [5], K-SVD [1], and more recent, scalable variants [14]. This dominant approach to dictionary learning exploits the fact that the constraint $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ is bilinear. Because the problem is nonconvex, spurious local minima are a concern in practice, and even in the cases where the algorithms perform well empirically, providing global theoretical guarantees would be a daunting task. Even the local properties of the problem have only recently begun to be studied carefully. For example, [10; 7] have shown that under certain natural random models for $\boldsymbol{X}$!, the desired solution will be a local minimum of the objective function with high probability. However, these results do not guarantee correct recovery by any efficient algorithm.

In this work, we contribute to the understanding of both of these questions in the case when $\boldsymbol{A}$ is square and nonsingular. We prove that $O(n \log n)$ samples are sufficient to uniquely determine the decomposition with high probability, under the assumption $\boldsymbol{X}$ is generated by a Bernoulli-Gaussian or Bernoulli-Rademacher process.

Our argument for uniqueness suggests a new, efficient dictionary learning algorithm, which we call Exact Recovery of Sparsely-Used Dictionaries (ER-SpUD). This algorithm solves a sequence of linear programs with varying constraints. We prove that under the aforementioned assumptions, the algorithm exactly recovers $\boldsymbol{A}$ and $\boldsymbol{X}$ with high probability. This result holds when the expected number of nonzero elements in each column of $\boldsymbol{X}$ is at most $O(\sqrt{n})$ and the number of samples $p$ is at least $\Omega(n^2 \log^2 n)$. To the best of our knowledge, this result is the first to demonstrate an efficient algorithm for dictionary learning with provable guarantees.

Moreover, we prove that this result is tight to within a $\log$ factor: for the Bernoulli-Gaussian case, when the expected number of nonzeros in each column is $\Omega(\sqrt{n \log n})$, algorithms of this style fail with high probability.

Our algorithm is related to previous proposals by Zibulevsky and Pearlmutter [22] (for source separation) and Gottlieb and Neylon [9] (for dictionary learning), but involves several new techniques that seem to be important for obtaining provable correct recovery – in particular, the use of sample vectors in the constraints. We will describe these differences more clearly in Section 5, after introducing our approach. Other related recent proposals include [17; 11].

The remainder of this paper is organized as follows. In Section 3, we fix our model. Section 4 discusses situations in which this problem is well-posed. Building on the intuition developed in this section, Section 5 introduces the ER-SpUD algorithm for dictionary recovery. In Section 6, we introduce our main theoretical results, which characterize the regime in which ER-SpUD performs correctly. Section 7 describes the key steps in our analysis. Technical lemmas and proofs are sketched; for full details please see the full version. Finally, in Section 8 we perform experiments corroborating our theory and suggesting the utility of our approach.

[1] Of course, for som! e applications, weaker notions than uniqueness may be of interest. For example, Vainsencher et. al. [19] give generalization bounds for a learned dictionary $\hat{\boldsymbol{A}}$. Compared to the results mentioned above, these bounds depend much more gracefully on the dimension and sparsity level. However, they do not directly imply that the "true" dictionary $\boldsymbol{A}$ is unique, or that it can be recovered by an efficient algorithm.

## 2. Notation

We write $\|\boldsymbol{v}\|_p$ for the standard $\ell^p$ norm of a vector $\boldsymbol{v}$, and we write $\|\boldsymbol{M}\|_p$ for the induced operator norm on a matrix $\boldsymbol{M}$. $\|\boldsymbol{v}\|_0$ denotes the number of non-zero entries in $\boldsymbol{v}$. We denote the Hadamard (point-wise) product by $\odot$. $[n]$ denotes the first $n$ positive integers, $\{1, 2, \ldots, n\}$. For a set of indices $I$, we let $\boldsymbol{P}_I$ denote the projection matrix onto the subspace of vectors supported on indices $I$, zeroing out the other coordinates. For a matrix $\boldsymbol{X}$ and a set of indices $J$, we let $\boldsymbol{X}_J$ ($\boldsymbol{X}^J$) denote the submatrix containing just the rows (columns) indexed by $J$. We write the standard basis vector that is non-zero in coordinate $i$ as $\boldsymbol{e}_i$. For a matrix $\boldsymbol{X}$ we let $\mathrm{row}(\boldsymbol{X})$ denote the span of its rows. For a set $S$, $|S|$ is its cardinality.

## 3. The Probabilistic Models

We analyze the dictionary learning problem under the assumption that $\boldsymbol{A}$ is an arbitrary nonsingular $n$-by-$n$ matrix, but that $\boldsymbol{X}$ is a random sparse $n$-by-$p$ matrix with i.i.d. entries. In the Bernoulli($\theta$)-Gaussian model, the entries of $\boldsymbol{X}$ are independent random variables, each of which has the form $X_{i,j} = \varsigma\tau$, where $\varsigma \sim N(0, 1)$ is a standard Gaussian, and $\tau$ is 1 with probability $\theta$ and 0 with probability $1 - \theta$, independent of $\varsigma$. We also consider a Bernoulli($\theta$)-Rademacher model, in which the non-zero entries are chosen uniformly in $\pm 1$.

## 4. When is the Factorization Unique?

At first glance, it seems the number of samples $p$ required to identify $\boldsymbol{A}$ could be quite large. For example, Aharon *et. al.* view the given data matrix $\boldsymbol{X}$ as having sparse columns, each with at most $k$ nonzero entries. If the given samples $\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j$ lie on an arrangement of $\binom{n}{k}$ $k$-dimensional subspaces $\mathrm{range}(\boldsymbol{A}_I)$, corresponding to possible support sets $I$, $\boldsymbol{A}$ is identifiable.

On the other hand, the most immediate lower bound on the number of samples required comes from the simple fact that to recover $\boldsymbol{A}$ we need to see at least one linear combination involving each of its columns. The "coupon collection" phenomenon tells us that $p = \Omega(\frac{1}{\theta} \log n)$ samples are required for this to occur with constant probability, where $\theta$ is the probability that an element $\boldsymbol{X}_{ij}$ is nonzero. When $\theta$ is as small as $O(1/n)$, this means $p$ must be at least proportional to $n \log n$. Our next result shows that, in fact, this lower bound is tight – the problem becomes well-posed once we have observed $cn \log n$ samples.

**Theorem 1 (Uniqueness)** *Under the Bernoulli($\theta$)-Gaussian and Bernoulli($\theta$)-Rademacher models, if $1/n \leq \theta \leq 1/C$ and $p > Cn \log n$, then with probability at least $1 - \exp\{-c'p\}$, for any alternative factorization $\boldsymbol{Y} = \boldsymbol{A}'\boldsymbol{X}'$ such that $\max_i \|\boldsymbol{e}_i^T \boldsymbol{X}'\|_0 \leq \max_i \|\boldsymbol{e}_i^T \boldsymbol{X}\|_0$, we have $\boldsymbol{A}' = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}$ and $\boldsymbol{X}' = \boldsymbol{\Lambda}^{-1}\boldsymbol{\Pi}^T \boldsymbol{X}$, for some permutation matrix $\boldsymbol{\Pi}$ and nonsingular diagonal matrix $\boldsymbol{\Lambda}$, for some absolute constants $C$ and $c'$.*

### 4.1. Sketch of Proof

Rather than looking at the problem as one of trying to recover the sparse columns of $\boldsymbol{X}$, we instead try to recover the sparse rows. As $\boldsymbol{X}$ is non-singular with very high probability, the following lemma tells us that for any other factorization the row spaces of $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{X}'$ are probably the same.

**Lemma 2** *If* $\operatorname{rank}(\boldsymbol{X}) = n$*,* $\boldsymbol{A}$ *is nonsingular, and* $\boldsymbol{Y}$ *can be decomposed into* $\boldsymbol{Y} = \boldsymbol{A}'\boldsymbol{X}'$*, then the row spaces of* $\boldsymbol{X}'$*,* $\boldsymbol{X}$*, and* $\boldsymbol{Y}$ *are the same.*

We will prove that the sparsest vectors in the row-span of $\boldsymbol{Y}$ are the rows of $\boldsymbol{X}$. As any other factorization $\boldsymbol{Y} = \boldsymbol{A}'\boldsymbol{X}'$ will have the same row-span, all of the rows of $\boldsymbol{X}'$ will lie in the row-span of $\boldsymbol{Y}$. This will tell us that they can only be sparse if they are in fact rows of $\boldsymbol{X}$. This is reasonable, since if distinct rows of $\boldsymbol{X}$ have nearly disjoint patterns of nonzeros, taking linear combinations of them will increase the number of nonzero entries.

**Lemma 3** *Let* $\boldsymbol{\Omega}$ *be an* $n$*-by-*$p$ *Bernoulli(*$\theta$*) matrix with* $1/n < \theta < 1/4$*. For each set* $S \subseteq [n]$*, let* $T_S \subseteq [p]$ *be the indices of the columns of* $\boldsymbol{\Omega}$ *that have at least one non-zero entry in some row indexed by* $S$*.*

*a. For every set* $S$ *of size* $2$*,*

$$\mathbb{P}\left[\, |T_S| \leq (4/3)\theta p \,\right] \leq \exp\left(-\theta p/108\right).$$

*b. For every set* $S$ *of size* $\sigma$ *with* $3 \leq \sigma \leq 1/\theta$

$$\mathbb{P}\left[\, |T_S| \leq (3\sigma/8)\theta p \,\right] \leq \exp\left(-\sigma\theta p/64\right).$$

*c. For every set* $S$ *of size* $\sigma$ *with* $1/\theta \leq \sigma$*,*

$$\mathbb{P}\left[\, |T_S| \leq (1 - 1/e)p/2 \,\right] \leq \exp\left(-(1 - 1/e)p/8\right).$$

Lemma 3 says that every subset of at least two rows of $\boldsymbol{X}$ is likely to be supported on many more than $\theta p$ columns, which is larger than the expected number of nonzeros $\theta p$ in rows of $\boldsymbol{X}$. We show that for any vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ with support $S$ of size at least 2, it is unlikely that $\boldsymbol{\alpha}^T\boldsymbol{X}$ is supported on many fewer columns than are in $T_S$.

**Lemma 4** *If* $\boldsymbol{X} = \boldsymbol{\Omega} \odot \boldsymbol{R}$ *for a binary matrix* $\boldsymbol{\Omega}$ *and an i.i.d. Gaussian matrix* $\boldsymbol{R}$*, then the probability that there is a vector* $\boldsymbol{\alpha}$ *with support* $S$ *such that*

$$\left\|\boldsymbol{\alpha}^T\boldsymbol{X}\right\|_0 \leq |T_S| - |S|$$

*is zero.*

In the next lemma, we call a vector $\boldsymbol{\alpha}$ fully dense if all of its entries are nonzero.

**Lemma 5** *For* $t > 200s$*, let* $\boldsymbol{\Omega} \in \{0,1\}^{s \times t}$ *be any binary matrix with at least one nonzero in each column. Let* $\boldsymbol{R}$ *be an* $s$*-by-*$t$ *matrix with Rademacher random entries, and let* $\boldsymbol{U} = \boldsymbol{\Omega} \odot \boldsymbol{R}$*. Then, the probability that there exists a fully-dense vector* $\boldsymbol{\alpha}$ *for which* $\left\|\boldsymbol{\alpha}^T\boldsymbol{U}\right\|_0 \leq t/5$ *is at most*

$$2^{-t/25}.$$

Combining Lemmas 3, 4 and 5, we prove the following.

**Lemma 6** *If $\boldsymbol{X}$ is an $n$-by-$p$ Bernoulli($\theta$)–Gaussian or Bernoulli($\theta$)–Rademacher matrix with $1/n < \theta < 1/C$ and $p > Cn \log n$ for a sufficiently large constant $C$, then the probability that there is a vector $\boldsymbol{\alpha}$ with support of size larger than $1$ for which*

$$\left\|\boldsymbol{\alpha}^T \boldsymbol{X}\right\|_0 \leq (11/9)\theta p$$

*is at most $\exp(-c\theta p)$, for some constant $c$.*

For convenience, this lemma is proved as Lemmas 16 and 17 in the Appendix. Theorem 1 follows from Lemmas 2 and 6.

## 5. Exact Recovery

Theorem 1 suggests that we can recover $\boldsymbol{X}$ by looking for sparse vectors in the row space of $\boldsymbol{Y}$. Any vector in this space can be generated by taking a linear combination $\boldsymbol{w}^T\boldsymbol{Y}$ of the rows of $\boldsymbol{Y}$ (here, $\boldsymbol{w}^T$ denotes the vector transpose). We arrive at the optimization problem

$$\text{minimize } \|\boldsymbol{w}^T\boldsymbol{Y}\|_0 \quad \text{subject to} \quad \boldsymbol{w} \neq \boldsymbol{0}.$$

Theorem 1 implies that any solution to this problem must satisfy $\boldsymbol{w}^T\boldsymbol{Y} = \lambda \boldsymbol{e}_j^T \boldsymbol{X}$ for some $j \in [n]$, $\lambda \neq 0$. Unfortunately, both the objective and constraint are nonconvex. We therefore replace the $\ell^0$ norm with its convex envelope, the $\ell^1$ norm, and prevent $\boldsymbol{w}$ from being the zero vector by constraining it to lie in an affine hyperplane $\{\boldsymbol{r}^T\boldsymbol{w} = 1\}$. This gives a linear programming problem of the form

$$\text{minimize } \|\boldsymbol{w}^T\boldsymbol{Y}\|_1 \quad \text{subject to} \quad \boldsymbol{r}^T\boldsymbol{w} = 1. \tag{1}$$

We will prove that this linear program is likely to produce rows of $\boldsymbol{X}$ when we choose $\boldsymbol{r}$ to be a column or a sum of two columns of $\boldsymbol{Y}$.

### 5.1. Intuition

To gain more insight into the optimization problem (1), we consider for analysis an equivalent problem, under the change of variables $\boldsymbol{z} = \boldsymbol{A}^T\boldsymbol{w}$, $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{r}$:

$$\text{minimize } \|\boldsymbol{z}^T\boldsymbol{X}\|_1 \quad \text{subject to} \quad \boldsymbol{b}^T\boldsymbol{z} = 1. \tag{2}$$

When we choose $\boldsymbol{r}$ to be a column of $\boldsymbol{Y}$, $\boldsymbol{b}$ becomes a column of $\boldsymbol{X}$. While we do not know $\boldsymbol{A}$ or $\boldsymbol{X}$ and so cannot directly solve problem (2), it is equivalent to problem (1): (1) recovers a row of $\boldsymbol{X}$ if and only if the solution to (2) is a scaled multiple of a standard basis vector: $\boldsymbol{z}_\star = \lambda \boldsymbol{e}_j$, for some $j$, $\lambda$.

To get some insight into why this might occur, consider what would happen if $\boldsymbol{X}$ exactly preserved the $\ell_1$ norm: i.e., if $\|\boldsymbol{z}^T\boldsymbol{X}\|_1 = c\|\boldsymbol{z}\|_1$ for all $\boldsymbol{z}$ for some constant $c$. The solution to (2) would just be the vector $\boldsymbol{z}$ of smallest $\ell^1$ norm satisfying $\boldsymbol{b}^T\boldsymbol{z} = 1$, which would be $\boldsymbol{e}_{j_\star}/b_{j_\star}$, where $j_\star$ is the index of the element of $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{r}$ of largest magnitude. The algorithm would simply extract the row of $\boldsymbol{X}$ that is most "preferred" by $\boldsymbol{b}$!

Under the random coefficient models considered here, $\boldsymbol{X}$ *approximately* preserves the $\ell_1$ norm, but does not exactly preserve it [15]. Our algorithm can tolerate this approximation if the largest element of $\boldsymbol{b}$ is significantly larger than the other elements. In this case we can still apply the above argument to show that (2) will recover the $j_\star$-th row of $\boldsymbol{X}$. In particular, if we let $|\boldsymbol{b}|_{(1)} \geq |\boldsymbol{b}|_{(2)} \geq \cdots \geq |\boldsymbol{b}|_{(n)}$ be the absolute values of the entries of $\boldsymbol{b}$ in decreasing order, we will require both $|\boldsymbol{b}|_{(2)}/|\boldsymbol{b}|_{(1)} < 1 - c/\log(n)$ and that the total number of nonzeros in $\boldsymbol{b}$ is at most $c/\theta$.

In the Bernoulli-Gaussian case, when we choose $\boldsymbol{r}$ to be a column of $\boldsymbol{Y}$ and thus $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{r}$ to be a column of $\boldsymbol{X}$, properties of the order statistics of Gaussian random vectors imply that our requirements are probably met. In the Bernoulli-Rademacher case all the non-zero entries of a column of $\boldsymbol{X}$ are $1$ or $-1$, and so there is no gap between the magnitudes of the largest and second-largest elements. For this reason, we choose $\boldsymbol{r}$ to be the sum of two columns of $\boldsymbol{Y}$ and thus $\boldsymbol{b}$ to be the sum of two columns of $\boldsymbol{X}$. When $\theta < 1/\sqrt{n}$, there is a reasonable chance that the support of these two columns overlap in exactly one element, in which case we obtain a gap between the magnitudes of the largest two elements in the sum. This modification also provides improvements in the Bernoulli-Gaussian model.

### 5.2. The Algorithms

Our algorithms are divided into two stages. In the first stage, we collect many potential rows of $\boldsymbol{X}$ by solving problems of the form (1). In the simpler Algorithm **ER-SpUD(SC)** ("single column"), we do this by using each column of $\boldsymbol{Y}$ as the constraint vector $\boldsymbol{r}$ in the optimization. In the slightly better Algorithm **ER-SpUD(DC)** ("double column"), we pair up all the columns of $\boldsymbol{Y}$ and then substitue the sum of each pair for $\boldsymbol{r}$. In the second stage, we use a greedy algorithm (Algorithm **Greedy**) to select a subset of $n$ of the rows produced. In particular, we choose a linearly independent subset among those with the fewest non-zero elements. From the proof of the uniqueness of the decomposition, we know with high probability that the rows of $\boldsymbol{X}$ are the sparsest $n$ vectors in $\text{row}(\boldsymbol{Y})$. Moreover, for $p \geq \Omega(n \log n)$, Theorems 7 and 8, along with the coupon collection phenomenon, tell us that a scaled multiple of each of the rows of $\boldsymbol{X}$ is returned by the first phase of our algorithm, with high probability.

```
ER-SpUD(SC): Exact Recovery of Sparsely-Used Dictionaries using single
columns of $\boldsymbol{Y}$ as constraint vectors.

    For $j = 1 \dots p$
        Solve $\min_{\boldsymbol{w}}\ \|\boldsymbol{w}^T\boldsymbol{Y}\|_1$ subject to $(\boldsymbol{Y}\boldsymbol{e}_j)^T\boldsymbol{w} = 1$, and set $\boldsymbol{s}_j = \boldsymbol{w}^T\boldsymbol{Y}$.
```

[2]

[2] Preconditioning by setting $\boldsymbol{Y}_p = (\boldsymbol{Y}\boldsymbol{Y}^T)^{-1/2}\boldsymbol{Y}$ helps in simulation, while our analysis does not require $\boldsymbol{A}$ to be well conditioned.

```
ER-SpUD(DC): Exact Recovery of Sparsely-Used Dictionaries using the sum of two columns of $Y$ as constraint vectors.
  1. Randomly pair the columns of $\boldsymbol{Y}$ into $p/2$ groups $g_j = \{\boldsymbol{Y}\boldsymbol{e}_{j_1}, \boldsymbol{Y}\boldsymbol{e}_{j_2}\}$.
  2. For $j = 1 \dots p/2$
         Let $\boldsymbol{r}_j = \boldsymbol{Y}\boldsymbol{e}_{j_1} + \boldsymbol{Y}\boldsymbol{e}_{j_2}$, where $g_j = \{\boldsymbol{Y}\boldsymbol{e}_{j_1}, \boldsymbol{Y}\boldsymbol{e}_{j_2}\}$.
         Solve $\min_{\boldsymbol{w}} \ \|\boldsymbol{w}^T\boldsymbol{Y}\|_1$ subject to $\boldsymbol{r}_j^T\boldsymbol{w} = 1$, and set $\boldsymbol{s}_j = \boldsymbol{w}^T\boldsymbol{Y}$.
```

```
Greedy: A Greedy Algorithm to Reconstruct $\boldsymbol{X}$ and $\boldsymbol{A}$.
  1. REQUIRE: $\mathcal{S} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_T\} \subset \mathbb{R}^p$.
  2. For $i = 1 \dots n$
         REPEAT
             $l \leftarrow \arg\min_{\boldsymbol{s}_l \in \mathcal{S}} \|\boldsymbol{s}_l\|_0$, breaking ties arbitrarily
             $\boldsymbol{x}_i = \boldsymbol{s}_l$
             $\mathcal{S} = \mathcal{S} \backslash \{\boldsymbol{s}_l\}$
         UNTIL rank([$\boldsymbol{x}_1, \dots, \boldsymbol{x}_i$])$= i$
  3. Set $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]^T$, and $\boldsymbol{A} = \boldsymbol{Y}\boldsymbol{Y}^T(\boldsymbol{X}\boldsymbol{Y}^T)^{-1}$.
```

**Comparison to Previous Work.** The idea of seeking the rows of $\boldsymbol{X}$ sequentially, by looking for sparse vectors in $\mathrm{row}(\boldsymbol{Y})$, is not new *per se*. For example, in [22], Zibulevsky and Pearlmutter suggested solving a sequence of optimization problems of the form

$$\text{minimize} \quad \|\boldsymbol{w}^T\boldsymbol{Y}\|_1 \quad \text{subject to} \quad \|\boldsymbol{w}\|_2^2 \geq 1.$$

However, the non-convex constraint in this problem makes it difficult to solve. In more recent work, Gottlieb and Neylon [9] suggested using linear constraints as in (1), but choosing $\boldsymbol{r}$ from the standard basis vectors $\boldsymbol{e}_1 \dots \boldsymbol{e}_n$.

The difference between our algorithm and that of Gottlieb and Neylon—the use of columns of the sample matrix $\boldsymbol{Y}$ as linear constraints instead of elementary unit vectors, is crucial to the functioning of our algorithm (simulations of their Sparsest Independent Vector algorithm are reported below). In fact, there are simple examples of orthonormal matrices $\boldsymbol{A}$ for which the algorithm of [9] provably fails, whereas Algorithm **ER-SpUD(SC)** succeeds with high probability. One concrete example of this is a Hadamard matrix: in this case, the entries of $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{e}_j$ all have exactly the same magnitude, and [9] fails because the gap between $|\boldsymbol{b}|_{(1)}$ and $|\boldsymbol{b}|_{(2)}$ is zero when $\boldsymbol{r}$ is chosen to be an elementary unit vector. In this situation, Algorithm **ER-SpUD(DC)** still succeeds with high probability.

## 6. Main Theoretical Results

The intuitive explanations in the previous section can be made rigorous. In particular, under our random models, we can prove that when the number of samples is reasonably large compared to the dimension, (say $p \sim n^2 \log^2 n$), with high probability in $\boldsymbol{X}$ the algorithm will succeed. We conjecture it is possible to decrease the dependency on $p$ to $O(n \log n)$.

**Theorem 7 (Correct recovery (single-column))** *Suppose $\boldsymbol{X}$ is Bernoulli($\theta$)-Gaussian. For some positive constants $\alpha$, $c_1$, and $n_0$, for all $n > n_0$, and for $p > c_1 n^2 \log^2 n$, if*

$$\frac{2}{n} \;\le\; \theta \;\le\; \frac{\alpha}{\sqrt{n}\log n}, \tag{3}$$

*then, with an exponentially small probability of failure, the Algorithm **ER-SpUD(SC)** recovers all $n$ rows of $\boldsymbol{X}$. That is, all $n$ rows of $\boldsymbol{X}$ are included in the $p$ potential vectors $\boldsymbol{w}_1^T \boldsymbol{Y}, \ldots, \boldsymbol{w}_p^T \boldsymbol{Y}$.*

The upper bound of $\alpha/\sqrt{n}\log n$ on $\theta$ has two sources: an upper bound of $\alpha/\sqrt{n}$ is imposed by the requirement that $\boldsymbol{b}$ be sparse. An additional factor of $\log n$ comes from the need for a gap between $|\boldsymbol{b}|_{(1)}$ and $|\boldsymbol{b}|_{(2)}$ of the $k$ i.i.d. Gaussian random variables. On the other hand, using the sum of two columns of $\boldsymbol{Y}$ as $\boldsymbol{r}$ can save the factor of $\log n$ in the requirement on $\theta$ since the "collision" of non-zero entries in the two columns of $\boldsymbol{X}$ creates a larger gap between $|\boldsymbol{b}|_{(1)}$ and $|\boldsymbol{b}|_{(2)}$. More importantly, the resulting algorithm is less dependent on the magnitudes of the nonzero elements in $\boldsymbol{X}$. The algorithm using a single column exploited the fact that there exists a reasonable gap between $|b|_{(1)}$ and $|b|_{(2)}$, whereas the two-column variant **ER-SpUD(DC)** succeeds even if the nonzeros all have the same magnitude.

**Theorem 8 (Correct recovery (two-column))** *Suppose $\boldsymbol{X}$ is Bernoulli($\theta$)-Gaussian or Bernoulli($\theta$)-Rademacher. For some $\alpha > 0$ and for all $n$ larger than some $n_0$, and $p > c_1 n^2 \log^2 n$, if the probability of non-zero entries $\theta$ satisfies*

$$\frac{2}{n} \le \theta \;\le\; \frac{\alpha}{\sqrt{n}}. \tag{4}$$

*Then with overwhelming probability, the Algorithm **ER-SpUD(DC)** recovers all $n$ rows of $\boldsymbol{X}$. That is, all $n$ rows of $\boldsymbol{X}$ are included in the $p/2$ potential vectors $\boldsymbol{w}_1^T \boldsymbol{Y}, \ldots, \boldsymbol{w}_{p/2}^T \boldsymbol{Y}$.*

Hence, as we choose $p$ to grow faster than $n^2 \log^2 n$, the algorithm will succeed with probability approaching one. That the algorithm succeeds is interesting, perhaps even unexpected. There is potentially a great deal of symmetry in the problem – all of the rows of $\boldsymbol{X}$ might have similar $\ell^1$-norm. The vectors $\boldsymbol{r}$ break this symmetry, preferring one particular solution at each step, at least in the regime where $\boldsymbol{X}$ is sparse. To be precise, the expected number of nonzero entries in each column must be bounded by $\sqrt{n \log n}$.

It is natural to wonder whether this is an artifact of the analysis, or whether such a bound is necessary. We can prove that for Algorithm **ER-SpUD(DC)**, the sparsity demands in Theorem 8 cannot be improved by more than a factor of $\sqrt{\log n}$. Consider the optimization problem (2). One can show that for each $i$, $\|\boldsymbol{e}_i^T \boldsymbol{X}\|_1 \approx \theta p$. Hence, if we set $\boldsymbol{z} = \boldsymbol{e}_{j_\star}/b_{j_\star}$, where $j_\star$ is the index of the largest element of $\boldsymbol{b}$ in magnitude, then

$$\|\boldsymbol{z}^T \boldsymbol{X}\|_1 \;=\; \frac{\|\boldsymbol{e}_{j_\star}^T \boldsymbol{X}\|_1}{\|\boldsymbol{b}\|_\infty} \;\approx\; C \frac{\theta p}{\sqrt{\log n}}.$$

If we consider the alternative solution $\boldsymbol{v} = \operatorname{sign}(\boldsymbol{b})/\|\boldsymbol{b}\|_1$, a calculation shows that

$$\|\boldsymbol{v}^T\boldsymbol{X}\|_1 \;\approx\; C'p/\sqrt{n}.$$

Hence, if $\theta > c\sqrt{\log n/n}$ for sufficiently large $c$, the second solution will have smaller objective function. These calculations are carried through rigorously in the full version, giving:

**Theorem 9** *For any fixed $\beta$ and sufficiently large $n$, and $p \geq C(\beta)n$, the following occurs. If the probability of nonzeros $\theta$ satisfies*

$$\theta \geq \sqrt{\frac{\beta \log n}{n}}, \tag{5}$$

*then the probability (in $\boldsymbol{X}$) that solving the optimization problem (1) with $\boldsymbol{r} = \boldsymbol{Y}\boldsymbol{e}_i$ or $\boldsymbol{r} = \boldsymbol{Y}\boldsymbol{e}_i + \boldsymbol{Y}\boldsymbol{e}_j$ recovers one of the rows of $\boldsymbol{X}$ is at most $n^{-c(\beta)}$, where $c(\beta) > 0$.*

This implies that the result in Theorem 7 is nearly the best possible for this algorithm, at least in terms of its demands on $\theta$.

## 7. Sketch of the Analysis

In this section, we sketch the arguments used to prove Theorem 7. The proof of Theorem 8 is similar. These arguments are carried through rigorously in the full version. At a high level, our argument follows the intuition of Section 5, using the order statistics and the sparsity property of $\boldsymbol{b}$ to argue that the solution must recover a row of $\boldsymbol{X}$. We say that a vector is $k$-sparse if it has at most $k$ non-zero entries. Our goal is to show that $\boldsymbol{z}_\star$ is 1-sparse. We find it convenient to do this in two steps.

We first argue that the solution $\boldsymbol{z}_\star$ to (2) must be supported on indices that are non-zero in $\boldsymbol{b}$, so $\boldsymbol{z}$ is at least as sparse as $\boldsymbol{b}$, say $\sqrt{n}$-sparse in our case. Using this result, we restrict our attention to a submatrix of $\sqrt{n}$ rows of $\boldsymbol{X}$, and prove that for this restricted problem, when the gap $1-|\boldsymbol{b}|_{(2)}/|\boldsymbol{b}|_{(1)}$ is large enough, the solution $\boldsymbol{z}_\star$ is in fact 1-sparse, and we recover a row of $\boldsymbol{X}$.

**Proof solution is sparse.** We first show that the solution $\boldsymbol{z}_\star$ to (2) is probably supported only on the non-zero indices in $\boldsymbol{b}$. Let $J$ denote the indices of the $s$ non-zero entries of $|\boldsymbol{b}|$, and let $S = \{j \mid \boldsymbol{X}_{J,j} \neq \boldsymbol{0}\} \subset [p]$, i.e., the indices of the nonzero columns in $X_J$, and write $\boldsymbol{z}_0 = \boldsymbol{P}_J\boldsymbol{z}_\star$ and $\boldsymbol{z}_1 = \boldsymbol{z}_\star - \boldsymbol{z}_0$. By definition, $\boldsymbol{z}_0$ is supported on $J$ and $\boldsymbol{z}_1$ on $J^c$. Moreover, $\boldsymbol{z}_0$ is feasible for Problem (2). We will show that it has at least as low an objective function value as $\boldsymbol{z}_\star$, and thus conclude that $\boldsymbol{z}_1$ must be zero. Write

$$\begin{aligned}\|\boldsymbol{z}_\star^T\boldsymbol{X}\|_1 \;&=\; \|\boldsymbol{z}_\star^T\boldsymbol{X}^S\|_1 + \|\boldsymbol{z}_\star^T\boldsymbol{X}^{S^c}\|_1 \;\geq\; \|\boldsymbol{z}_0^T\boldsymbol{X}^S\|_1 - \|\boldsymbol{z}_1^T\boldsymbol{X}^S\|_1 + \|\boldsymbol{z}_1^T\boldsymbol{X}^{S^c}\|_1 \\ &= \|\boldsymbol{z}_0^T\boldsymbol{X}\|_1 - 2\|\boldsymbol{z}_1^T\boldsymbol{X}^S\|_1 + \|\boldsymbol{z}_1^T\boldsymbol{X}\|_1,\end{aligned} \tag{6}$$

where we have used the triangle inequality and the fact that $\boldsymbol{z}_0^T\boldsymbol{X}^{S^c} = \boldsymbol{0}$. In expectation we have that

$$\|\boldsymbol{z}_\star^T\boldsymbol{X}\|_1 \;\geq\; \|\boldsymbol{z}_0^T\boldsymbol{X}\|_1 + (p-2|S|)\mathbb{E}[\|\boldsymbol{z}_1^T\boldsymbol{X}\|_1] \;\geq\; \|\boldsymbol{z}_0^T\boldsymbol{X}\|_1 + c(p-2|S|)\sqrt{\theta/n}\|\boldsymbol{z}_1\|_1, \tag{7}$$

where the last inequality requires $\theta n \geq 2$.

So as long as $p - 2|S| > 0$, $\boldsymbol{z}_0$ has lower expected objective value. To prove that this happens with high probability, we first upper bound $|S|$ by the number of nonzeros in $X_J$, which in expectation is $\theta sp$. As long as $p - 2(1+\delta)\theta sp = p(1 - c'\theta s) > 0$, or equivalently $s < c_s/\theta$ for some constant $c_2$, we have $\|\boldsymbol{z}_\star^T \boldsymbol{X}\|_1 > \|\boldsymbol{z}_0^T \boldsymbol{X}\|_1$. In the following lemma, we make this argument formal by proving concentration around the expectation.

**Lemma 10** *For some positive constants $\eta$, $c_1$, $c_2$ and $n_0$, if $2 < \theta n < \eta\sqrt{n}$, $n > n_0$, $p > c_1 n^2 \log^2 n$ and $\|b\|_0 = s < \frac{c_2}{\theta}$, then $z_\star$ is supported only on the non-zero indices of $b$ with probability tending to $1$ as $n$ goes to infinity.*

Note in problem (2), $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{r}$. If we choose $\boldsymbol{r} = \boldsymbol{Y}\boldsymbol{e}_i$, then $\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{Y}\boldsymbol{e}_i = \boldsymbol{X}\boldsymbol{e}_i$, and $\mathbb{E}[\|\boldsymbol{b}\|_0] = \theta n$. A Chernoff bound then tells us that with high probability $\boldsymbol{z}_\star$ is supported on no more than $2\theta n$ entries, i.e., $s < 2\theta n$. Thus as long as $2\theta n < c_2/\theta$, i.e., $\theta < c_\theta/\sqrt{n}$, we have $\|\boldsymbol{z}_\star\|_0 < 2\theta n = c_\theta\sqrt{n}$.

**The solution in $\boldsymbol{X}_J$:** If we restrict our attention to the induced $s$-by-$p$ submatrix $X_J$, we observe that $\boldsymbol{X}_J$ is incredibly sparse – most of the columns have at most one nonzero entry. Arguing as we did in the first step, let $j^\star$ denote the index of the largest entry of $|\boldsymbol{b}_J|$, and let $S = \{j \mid \boldsymbol{X}_J(j^\star, j) \neq \boldsymbol{0}\} \subset [p]$, i.e., the indices of the nonzero entries in the $j^\star$-th row of $X_J$. Without loss of generality, let's assume $b_{j^\star} = 1$. For any $\boldsymbol{z}$, write $\boldsymbol{z}_0 = \boldsymbol{P}_{j^\star}\boldsymbol{z}$ and $\boldsymbol{z}_1 = \boldsymbol{z} - \boldsymbol{z}_0$. Clearly $\boldsymbol{z}_0$ is supported on the $j^\star$-th entry and $\boldsymbol{z}_1$ on the rest. As in the first step,

$$\|\boldsymbol{z}^T \boldsymbol{X}_J\|_1 \geq \|\boldsymbol{z}_0^T \boldsymbol{X}_J\|_1 - 2\|\boldsymbol{z}_1^T \boldsymbol{X}_J^S\|_1 + \|\boldsymbol{z}_1^T \boldsymbol{X}_J\|_1, \tag{8}$$

By restricting our attention to 1-sparse columns of $\boldsymbol{X}_J$, we prove that with high probability

$$\|\boldsymbol{z}_1^T \boldsymbol{X}_J\|_1 \geq \sqrt{2/\pi}\theta p(1 - s\theta)(1-\varepsilon)^2\|\boldsymbol{z}_1\|_1.$$

We prove that with high probability the second term satisfies

$$\|\boldsymbol{z}_1^T \boldsymbol{X}_{J,S}\|_1 \leq (1+\epsilon)\sqrt{2/\pi}\theta^2 p\|\boldsymbol{z}_1\|_1.$$

For the first term, we show

$$\|\boldsymbol{z}_0^T \boldsymbol{X}_J\|_1 \geq \|\boldsymbol{e}_{j^\star}^T \boldsymbol{X}_J\|_1 - |\boldsymbol{b}_J^T \boldsymbol{z}_1|\|\boldsymbol{X}_J\|_1 \geq \|\boldsymbol{e}_{j^\star}^T \boldsymbol{X}_J\|_1 - |\boldsymbol{b}_J^T \boldsymbol{z}_1|(1+\epsilon)\sqrt{2/\pi}\theta p.$$

If $|\boldsymbol{b}|_{(2)}/|\boldsymbol{b}|_{(1)} < 1 - c/\log(n)$, then $|\boldsymbol{b}_J^T \boldsymbol{z}_1| \leq (1 - c/\log(n))\|\boldsymbol{z}_1\|_1$.

In Lemma 11, we combine these inequalities and choose the constants to show that if $\theta \leq c/\sqrt{n}\log n$, then

$$\|(\boldsymbol{z}_0 + \boldsymbol{z}_1)^T \boldsymbol{X}_J\|_1 \geq \|\boldsymbol{e}_{j^\star}^T \boldsymbol{X}_J\|_1 + \sqrt{\frac{2}{\pi}}\theta p(1 - \frac{c'}{\log n})\|\boldsymbol{z}_1\|_1. \tag{9}$$

Since $e_{j^\star}$ is a feasible solution to problem 2 with a lower objective value as long as $\boldsymbol{z}_1 \neq 0$, we know $e_{j^\star}$ is the only optimal solution. The following lemma makes this precise.

**Lemma 11** *Set $s \leq c_2\sqrt{n}$. If $\theta < \frac{c}{\sqrt{n}\log n}$, $n > n_0$, and $p > c_1 n^2 \log^2 n$, then with high probability the random matrix $\boldsymbol{X}$ has the following property:*

*For every* $J \in \binom{[n]}{s}$ *and every* $\boldsymbol{b} \in \mathbb{R}^s$ *satisfying* $|\boldsymbol{b}|_{(2)}/|\boldsymbol{b}|_{(1)} \leq 1 - c'/\log n$*, the solution to the restricted problem,*

$$\text{minimize } \|\boldsymbol{z}^T \boldsymbol{X}_J\|_1 \text{ s.t. } \boldsymbol{b}^T \boldsymbol{z} = 1, \tag{10}$$

*is unique and* $1$*-sparse.*

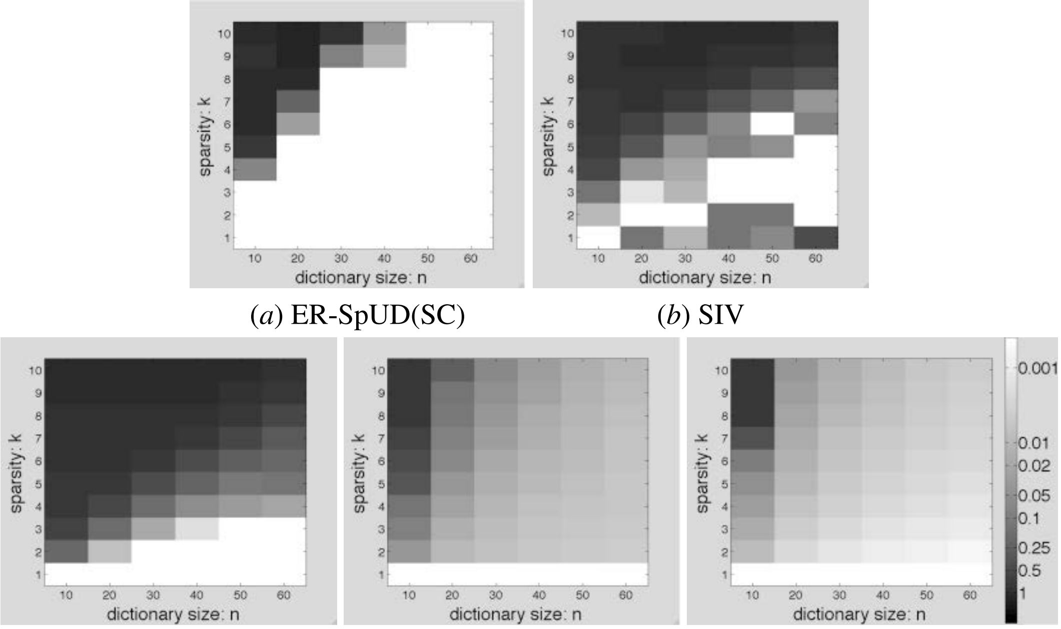
Once we know that a column of $\boldsymbol{Y}$ provides us with a constant probability of recovering one row of $\boldsymbol{X}$, we know that we need only use $O(n \log n)$ columns to recover all the rows of $\boldsymbol{X}$ with high probability. It turns out the dominant term of the failure probability is the one in Lemma 10.

## 8. Simulations

In this section we systematically evaluate our algorithm, and compare it with the state-of-the-art dictionary learning algorithms, including K-SVD [1], online dictionary learning [14], SIV [9], and the relative Newton method for source separation [21]. The first two methods are not limited to square dictionaries, while the final two methods, like ours, exploit properties of the square case. The method of [21] is similar in provenance to the incremental nonconvex approach of [22], but seeks to recover all of the rows of $\boldsymbol{X}$ simultaneously, by seeking a local minimum of a larger nonconvex problem. As our emphasis in this paper is mostly on correctness of the solution, we modify the default settings of these packages to obtain more accurate results (and hence a fairer comparison). For K-SVD, we use high accuracy mode, and switch the number of iterations from 10 to 30. Similarly, for relative Newton, we allow 1,000 iterations. For online dictionary learning, we allow 1,000. We observed diminishing returns beyond these numbers. Since K-SVD and online dictionary learning tend to get stuck at local optimum, for each trial we restart K-SVD and Online learning algorithm 5 times with randomized initializations and report the best performance. We measure accuracy in terms of the relative error, after permutation-scale ambiguity has been removed:

$$\tilde{\text{re}}(\hat{\boldsymbol{A}}, \boldsymbol{A}) \doteq \min_{\boldsymbol{\Pi}, \boldsymbol{\Lambda}} \|\hat{\boldsymbol{A}}\boldsymbol{\Lambda}\boldsymbol{\Pi} - \boldsymbol{A}\|_F / \|\boldsymbol{A}\|_F.$$

**Phase transition graph.** In our experiments we have chosen $\boldsymbol{A}$ to be a an $n$-by-$n$ matrix of independent Gaussian random variables. The coefficient matrix $\boldsymbol{X}$ is $n$-by-$p$, where $p = 5n \log_e n$. Each column of $\boldsymbol{X}$ has $k$ randomly chosen non-zero entries. In our experiments we have varied $n$ between 10 and 60 and $k$ between 1 and 10. Figure 1 shows the results for each method, with the average relative error reported in greyscale. White means zero error and black is 1. When $n$ is small, the relative Newton method appears to be able to handle a denser $\boldsymbol{X}$, while as $n$ grows large, ER-SpUD is more precise. In fact, empirically the phase transition between success and failure for ER-SpUD is quite sharp – problems below the boundary are solved to high numerical accuracy, while beyond the boundary the algorithm breaks down. In contrast, both online dictionary learning and relative Newton exhibit neither the same accuracy, nor the same sharp transition to failure – even in the black region of the graph, they still return solutions that are not completely wrong. The breakdown boundary of K-SVD is clear compared to online learning and relative Newton. As an active set algorithm, when it reaches a correct solution, the numerical accuracy is quite high. However, in our simulations we observe that both K-SVD and online learning may be trapped into a local optimum even for relatively sparse problems.



(*a*) ER-SpUD(SC) (*b*) SIV

(*c*) K-SVD (*d*) Online (*e*) Rel. Newton

Figure 1: Mean relative errors over 10 trials, with varying support $k$ (y-axis, increase from bottom to top) and basis size $n$(x-axis, increase from left to right). Here, $p = 5n \log_e n$. Our algorithm using a column of $Y$ as $r$ (ER-SpUD), SIV [9], K-SVD [1], online dictionary learning [14], and the relative Newton method for source separation [21].

## 9. Discussion

The main contribution of this work is a dictionary learning algorithm with provable performance guarantees under a random coefficient model. To our knowledge, this result is the first of its kind. However, it has two clear limitations: the algorithm requires that the reconstruction be exact, i.e., $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ and it requires $\boldsymbol{A}$ to be square. It would be interesting to address both of these issues (see also [2] for investigation in this direction). Finally, while our results pertain to a specific coefficient model, our analysis generalizes to other distributions. Seeking meaningful, deterministic assumptions on $\boldsymbol{X}$ that will allow correct recovery is another interesting direction for future work.

## Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant No. 0915487. JW also acknowledges support from Columbia University.

## References

[1] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322, 2006.

[2] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. Technical report, Technical report HAL-00345747, http://hal.archives-ouvertes.fr/hal-00354771/fr/, 2008.

[3] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[4] P. Comon. Independent component analysis: A new concept? *Signal Processing*, 36:287–314, 1994.

[5] K. Engan, S. Aase, and J. Hakon-Husoy. Method of optimal directions for frame design. In *ICASSP*, volume 5, pages 2443–2446, 1999.

[6] P. Erdös. On a lemma of Littlewood and Offord. *Bulletin of the American Mathematical Society*, 51:898–902, 1945.

[7] Q. Geng and J. Wright. On the local correctness of $\ell^1$ minimization for dictionary learning. *CoRR*, 2011.

[8] P. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4), 2005.

[9] L.-A. Gottlieb and T. Neylon. Matrix sparsication and the sparse null space problem. *APPROX and RANDOM*, 6302:205–218, 2010.

[10] R. Gribonval and K. Schnass. Dictionary identification-sparse matrix-factorisation via $l_1$-minimisation. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.

[11] F. Jaillet, R. Gribonval, M. Plumbley, and H. Zayyani. An l1 criterion for dictionary learning by subspace identification. In *IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5482–5485, 2010.

[12] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, and T. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(20):349–396, 2003.

[13] M. E. M. Aharon and A. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 416:48–67, 2006.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696, 2009.

[15] J. Matousek. On variants of the johnson-lindenstrauss lemma. *Wiley InterScience (www.interscience.wiley.com)*.

[16] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6538):607–609, 1996.

[17] M. Plumbley. Dictionary learning for $\ell^1$-exact sparse coding. In *Independent Component Analysis and Signal Separation*, pages 406–413, 2007.

[18] R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[19] D. Vainsencher, S. Mannor, and A. Bruckstein. The sample complexity of dictionary learning. In *Proc. Conference on Learning Theory*, 2011.

[20] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[21] M. Zibulevsky. Blind source separation with relative newton method. *Proceedings ICA*, pages 897–902, 2003.

[22] M. Zibulevsky and B. Pearlmutter. Blind source separation by sparse decomposition. *Neural Computation*, 13(4), 2001.

## Appendix A. Proof of Uniqueness

In this section we prove our upper bound on the number of samples for which the decomposition of $\boldsymbol{Y}$ into $\boldsymbol{A}\boldsymbol{X}$ with sparse $\boldsymbol{X}$ is unique up to scaling and permutation. We will consider $\boldsymbol{X}$ generated by both Bernoulli-Gaussian and Bernoulli-Rademacher processes. We will view $\boldsymbol{X}$ as the component-wise product of two matrices, $\boldsymbol{\Omega}$ and $\boldsymbol{R}$, and denote this product by $\boldsymbol{\Omega} \odot \boldsymbol{R}$, where

$$(\boldsymbol{\Omega} \odot \boldsymbol{R})\,(i,j) = \boldsymbol{\Omega}(i,j)\boldsymbol{R}(i,j).$$

We will let $\boldsymbol{\Omega}$ be an Bernoulli random matrix whose entries are $1$ with probability $\theta$ and zero otherwise. We will let $\boldsymbol{R}$ be a matrix of i.i.d. Gaussian or Rademacher random variables, as appropriate.

### A.1. Proof of Lemma 2

**Proof** Since $\operatorname{rank}(\boldsymbol{X}) = n$, we know

$$\operatorname{rank}(\boldsymbol{A}') \geq \operatorname{rank}(\boldsymbol{Y}) = \operatorname{rank}(\boldsymbol{A}) = n$$

Since both $\boldsymbol{A}$ and $\boldsymbol{A}'$ are nonsingular, the row spaces of $\boldsymbol{X}'$ and $\boldsymbol{X}$ are the same as that of $\boldsymbol{Y}$. ■

### A.2. Proof of Lemma 3

**Proof** First consider sets $S$ of two rows. The expected number of columns that have non-zero entries in at least one of these two rows is

$$p(1-(1-\theta)^2) = p(2\theta - \theta^2) \geq (3/2)p\theta,$$

for $\theta \leq 1/2$. Part $a$ now follows from a Chernoff bound.

For sets $S$ of size $\sigma \geq 3$, we divide our analysis into two cases. If $\sigma\theta < 1$, we observe that for every $S$

$$\begin{aligned}\mathbf{E}\,|T_S| &= p - (1-\theta)^\sigma p \\ &\geq (\sigma\theta - \tbinom{\sigma}{2}\theta^2)p \\ &= (1 - \frac{\sigma - 1}{2}\theta)\sigma\theta p \\ &\geq \frac{\sigma}{2}\theta p,\end{aligned}$$

where the inequalities follow from $\sigma\theta < 1$. Part $b$ now follows from a Chernoff bound.

If $\sigma\theta > 1$, for every $S$ of size $\sigma$ we have

$$\mathbf{E}\,|T_S| \geq (1 - e^{-\sigma\theta})p \geq (1 - e^{-1})p.$$

As before, part $c$ follows from a Chernoff bound. ■

### A.3. Proof of Lemma 4

We will now show that for every vector $\boldsymbol{\alpha}$ with support $S$, the number of non-zero entries in $\boldsymbol{\alpha}^T X$ is unlikely to be too much lower than the size of $T_S$.

The following definition and lemma are the key to our proof in the Bernoulli-Gaussian case.

**Definition 12 (fully dense vector)** *We call a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ fully dense if for all $i \in [n]$, $\alpha_i \neq 0$.*

**Lemma 13** *Let $\boldsymbol{\Omega} \in \{0,1\}^{n\times n}$ be any binary matrix with at least one nonzero in each column. Let $\boldsymbol{V} \sim_{iid} N(0,1)$, and set $\boldsymbol{U} = \boldsymbol{\Omega} \odot \boldsymbol{V}$. Then with probability one in the random matrix $\boldsymbol{V}$, the left nullspace of $\boldsymbol{U}$ does not contain any fully dense vector.*

**Proof** Let $\boldsymbol{U} = [\boldsymbol{u}_1 | \dots | \boldsymbol{u}_n]$ denote the columns of $\boldsymbol{U}$. For each $j \in [n]$, let $N_j$ be the left nullspace of $[\boldsymbol{u}_1 | \dots | \boldsymbol{u}_j]$, and let $N_0 \doteq \mathbb{R}^n$. Then

$$N_0 \supseteq N_1 \supseteq \cdots \supseteq N_n.$$

We need to show that $N_n$ does not contain a fully dense vector. If any $N_{j-1}$ does not contain a fully dense vector, we are done. On the other hand, suppose that $N_{j-1}$ contains a fully dense vector $\boldsymbol{\alpha}$. Fix any such fully dense $\boldsymbol{\alpha} \in N_{j-1}$. Since $u_j$ has some non-zero entry and it is independent of the columns $\boldsymbol{u}_1 \dots \boldsymbol{u}_{j-1}$, with probability one over the choice of $\boldsymbol{v}_j$, $\boldsymbol{\alpha}^T \boldsymbol{u}_j \neq 0$, and hence $\dim(N_j) \leq \dim(N_{j-1}) - 1$. Since $\dim(N_j) \geq \dim(N_{j-1}) - 1$, in this case $\dim(N_j) = \dim(N_{j-1}) - 1$. By induction, we may conclude that with probability 1 over the choice of $\boldsymbol{v}_1, \dots, \boldsymbol{v}_j$, either $N_j$ does not contain a dense vector, or $\dim(N_j) = n - j$. We conclude that either the left null space of $\boldsymbol{U}$ does not contain a dense vector, or its dimension is $0$. ■

**Proof of Lemma 4**

**Proof** Let $\boldsymbol{M}$ be the submatrix of $\boldsymbol{\Omega}$ containing the rows indexed by $S$ and the columns indexed by $T_S$. Let $\boldsymbol{\alpha}_S$ be the restriction of $\boldsymbol{\alpha}$ to the indices in $S$. As $S$ is the support of $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}_S$ is fully-dense.

Moreover, every column of $\boldsymbol{M}$ has at least one non-zero entry. If $\boldsymbol{\alpha}_S^T \boldsymbol{M}$ had $|T_S| - |S|$ zero entries, then $\boldsymbol{M}$ would have a square submatrix with $\boldsymbol{\alpha}_S$ in its nullspace. By Lemma 13, the probability that this happens is zero. ■

### A.4. Proof of Lemma 6

In the Bernoulli-Rademacher case, use the following theorem of Erdös.

**Theorem 14 ([6])** *For every $k \geq 2$ and real numbers $z_1, \ldots, z_k$,*

$$\mathbb{P}\left[\sum_i z_i r_i = 0\right] \leq 2^{-k}\binom{k}{\lfloor k/2 \rfloor} \leq 1/2,$$

*where each $r_i$ is chosen independently from $\pm 1$,*

**Lemma 15** *For $b > s$, let $\boldsymbol{\Omega} \in \{0,1\}^{s \times b}$ be any binary matrix with at least one nonzero in each column. Let $\boldsymbol{R}$ be an $s$-by-$b$ matrix with Rademacher random entries, and let $\boldsymbol{U} = \boldsymbol{\Omega} \odot \boldsymbol{R}$. Then, the probability that the left nullspace of $\boldsymbol{U}$ contains a fully dense vector is at most*

$$2^{-b+s\log(e^2 b/s)}$$

**Proof** As in the preceding lemma, we let $\boldsymbol{U} = [\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_b]$ denote the columns of $\boldsymbol{U}$ and for each $j \in [b]$, we let $N_j$ be the left nullspace of $[\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_j]$. We will show that it is very unlikely that $N_b$ contains a fully dense vector.

To this end, we show that if $N_{j-1}$ contains a fully dense vector, then with probability at least $1/2$ the dimension of $N_j$ is less than the dimension of $N_{j-1}$. To be concrete, assume that the first $j-1$ columns of $\boldsymbol{R}$ have been fixed and that $N_{j-1}$ contains a fully dense vector. Let $\boldsymbol{\alpha}$ be any such vector. If $\boldsymbol{u}_j$ contains only one non-zero entry, then $\boldsymbol{\alpha}^T \boldsymbol{u}_j \neq 0$ and so the dimension of $N_j$ is less than the dimension of $N_{j-1}$. If $\boldsymbol{u}_j$ contains more than one non-zero entry, each of its non-zero entries are random Rademacher random variables. So, Theorem 14 implies that the probability over the choice of entries in the $j$th column of $\boldsymbol{R}$ that $\boldsymbol{\alpha}^T \boldsymbol{u}_j = 0$ is at most one-half. So, with probability at least $1/2$ the dimension of $N_j$ is less than the dimension of $N_{j-1}$.

To finish the proof, we observe that the dimension of the nullspaces cannot decrease more than $s$ times. In particular, for $N_b$ to contain a fully dense vector, there must be at least $b - s$ columns for which the dimension of the nullspace does not decrease. Let $F \subset [b]$ have size $b - s$. The probability that for each $j \in F$ that $N_{j-1}$ contains a fully dense vector and that the dimension of $N_j$ equals the dimension of $N_{j-1}$ is at most $2^{-b+s-1}$. Taking a union bound over the choices for $F$, we see that the probability that $N_b$ contains a fully dense vector is at most

$$\binom{b}{b-s} 2^{-b+s} = \binom{b}{s} 2^{-b+s} \leq \left(\frac{eb}{s}\right)^s 2^{-b+s} \leq 2^{-b+s+s\log(eb/s)} = 2^{-b+s\log(e^2 b/s)}.$$

■

**Proof** [Proof of Lemma 5] If there is a fully-dense vector $\boldsymbol{\alpha}$ for which $\left\|\boldsymbol{\alpha}^T \boldsymbol{U}\right\|_0 \leq t/5$, then there is a subset of at least $b = 4t/5$ columns of $\boldsymbol{U}$ for which $\boldsymbol{\alpha}$ is in the nullspace of the restriction of

$\boldsymbol{U}$ to those columns. By Lemma 15, the probability that this happens for any particular subset of $b$ columns is at most

$$2^{-b+s\log e^2b/s} \leq 2^{-4t/5+s\log(e^2t/s)}.$$

Taking a union bound over the subsets of $b$ columns, we see that the probability that this can happen is at most

$$\binom{t}{4t/5} 2^{-4t/5+s\log e^2t/s} \leq 2^{0.722t}2^{-t(4/5-(s/t)\log(e^2t/s))} \leq 2^{t(0.722-0.8+0.0365)} \leq 2^{-t/25},$$

where in the first inequality we bound the binomial coefficient using the exponential of the corresponding binary entropy function, and in the second inequality we exploit $s/t < 1/200$. ■

**Lemma 16** *If $\boldsymbol{X}$ is an $n$-by-$p$ $\theta$-Bernoulli-Rademacher matrix with $1/n < \theta < 1/C$ and $p > Cn\log n$ for a sufficiently large constant $C$, then the probability that there is a vector $\boldsymbol{\alpha}$ with support of size larger than $1$ for which*

$$\left\|\boldsymbol{\alpha}^T\boldsymbol{X}\right\|_0 \leq (11/9)\theta p$$

*is at most*

$$\exp(-c\theta p),$$

*for some constant $c$.*

**Proof** Rather than considering vectors, we will consider the sets on which they are supported. So, let $S \subseteq [n]$ and let $\sigma = |S|$. We first consider the case when $17 \leq \sigma \leq 1/\theta$. Let $T$ be the set of columns of $\boldsymbol{X}$ that have non-zero entries in the rows indexed by $S$. Let $t = |T|$. By Lemma 3,

$$\mathbb{P}\left[t < (3/8)\sigma\theta p\right] \leq \exp(-\sigma\theta p/64).$$

Given that $t \geq (3/8)\sigma\theta p$, Lemma 5 tells us that the probability that there is a vector $\boldsymbol{\alpha}$ with support exactly $S$ for which

$$\left\|\boldsymbol{\alpha}^T\boldsymbol{X}\right\|_0 < (11/9)\theta p \leq (3/40)\sigma\theta p$$

is at most

$$\exp(-(3/200)\sigma\theta p).$$

Taking a union bound over all sets $S$ of size $\sigma$, we see that the probability that there vector $\boldsymbol{\alpha}$ of support size $\sigma$ such that $\left\|\boldsymbol{\alpha}^T\boldsymbol{X}\right\|_0 < (11/9)\theta p$ is at most

$$\binom{n}{\sigma}\left(\exp(-(3/200)\sigma\theta p) + \exp(-\sigma\theta p/64)\right) \leq \exp(-c\sigma\theta p),$$

for some constant $c$ given that $p > Cn\log n$ for a sufficiently large $C$.

For $\sigma \geq 1/\theta$, we may follow a similar argument to show that the probability that there is a vector $\boldsymbol{\alpha}$ with support size $\sigma$ for which $\left\|\boldsymbol{\alpha}^T\boldsymbol{X}\right\|_0 < (11/9)\theta p$ is at most

$$\exp(-cp),$$

for some other constant $c$. Summing these bounds over all $\sigma$ between 17 and $n$, we see that the probability that there exists a vector $\boldsymbol{\alpha}$ with support of size at least 17 such that such that $\left\|\boldsymbol{\alpha}^T \boldsymbol{X}\right\|_0 < (11/9)\theta p$ is at most

$$\exp(-c\theta p),$$

for some constant $c$.

To finish, we sketch a proof of how we handle the sets of support between 2 and 17. For $\sigma$ this small and for $\theta$ sufficiently small relative to $\sigma$ (that is smaller than some constant depending on $\sigma$), each of the columns in $T$ probably has exactly one non-zero entry. Again applying a Chernoff bound and a union bound over the choices of $S$, we can show that with probability $1 - \exp(-c\theta p)$ for every vector $\boldsymbol{\alpha}$ with support of size between 2 and 17, $\left\|\boldsymbol{\alpha}^T \boldsymbol{X}\right\|_0 \geq (5/4)\theta p$. ■

By a similar argument, we can prove the following Lemma for the Bernoulli-Gaussian case. The main difference in the proof is that we can use Lemma 4, and that we only need to treat the case of $\sigma = 2$ differently.

**Lemma 17** *If $\boldsymbol{X}$ is an $n$-by-$p$ $\theta$-Bernoulli-Gaussian matrix with $1/n < \theta < 1/4$ and $p > Cn \log n$ for a sufficiently large constant $C$, then the probability that there is a vector $\boldsymbol{\alpha}$ with support of size larger than $1$ for which*

$$\left\|\boldsymbol{\alpha}^T \boldsymbol{X}\right\|_0 \leq (11/9)\theta p$$

*is at most*

$$\exp(-c\theta p),$$

*for some constant $c$.*

### A.5. Proof of Theorem 1

We first observe that the rows of $\boldsymbol{X}$ are probably sparse.

**Lemma 18** *For $\boldsymbol{X}$ a $\theta$-Bernoulli-Gaussian or $\theta$-Bernoulli-Rademacher random matrix with $n$ rows and $p$ columns, the probability that any row of $\boldsymbol{X}$ has more than*

$$(10/9)\theta p$$

*non-zero entries is at most*

$$n \exp\{-\frac{\theta p}{243}\}.$$

**Proof** The expected number of non-zero entries in a row of $\boldsymbol{X}$ is $\theta p$. The lemma now follows from a Chernoff bound and a union bound over the $n$ rows. ■

**Proof of Theorem 1**

**Proof** From Lemmas 17 we know that the probability that $\boldsymbol{X}$ is singular is at most the above error probability. Given that $\boldsymbol{X}$ is non-singular, we know from Lemma 2 that the row-space of $\boldsymbol{Y}$ is the same as the row space of $\boldsymbol{X}$. So, it suffices to prove that the row space of $\boldsymbol{X}$ does not contain any vectors sparser than the rows of $\boldsymbol{X}$ itself. This follows from Lemma 18 and 17. ■