Bull Braz Math Soc, New Series (2020) 51:719–743
https://doi.org/10.1007/s00574-019-00171-y

Fast Overcomplete Dictionary Construction with Probabilistic Guarantees

Enrico Au-Yeung[1] · Greg Zanotti[1]

Received: 18 December 2018 / Accepted: 30 September 2019 / Published online: 9 October 2019

Abstract
In dictionary learning, a matrix comprised of signals Y is factorized into the product of two matrices: a matrix of prototypical atoms D, and a sparse matrix containing coefficients for atoms in D, called X. This process has applications in signal processing, image recognition, and a number of other fields. Many procedures for solving the dictionary learning problem follow the alternating minimization paradigm; that is, by alternating between solving for D and X, until the procedure converges to a solution. Our findings indicate that the costly step of alternating minimization can be avoided in some cases, by modifying an initialization procedure that was proposed in 2014. We provide theoretical justification and empirical evidence showing that atom recovery and reasonable data reconstruction is possible under these new assumptions.

Keywords Signal processing · Dictionary learning

1 Introduction and Motivation

Dictionary learning is an approach to characterize a large collection of data (commonly referred to as signals) by sparse linear combinations of a small set of prototypical signals. We can think of these prototypical signals as the representatives for the larger data set. This small set of representatives is known as the dictionary. Modeling signals with such sparse decompositions is very effective in many signal processing applications. For example, dictionary learning has been applied to perform face recognition (Zhang and Li 2010), image restoration and inpainting [even when the image is heavily corrupted (Mairal et al. (2008)) or data is limited or incomplete (Naumova and Schnass (2017))], and modeling ofdata with hierarchical structure, such as images and text

✉ Enrico Au-Yeung
eauyeun1@depaul.edu

Greg Zanotti
gzanotti@mail.depaul.edu

[1] Department of Mathematical Sciences, DePaul University, Chicago, IL 60614, USA

(Jenatton et al. 2010). Other examples, including medical imaging, can be found in the survey (Tosic and Frossard 2011).

Rather than using predefined bases, such as wavelets, that do not depend on the data, the goal of dictionary learning is to learn the dictionary from the sample data. Initially introduced by Olshausen and Field for modeling the spatial receptive fields of simple cells in the visual cortex, the idea of learning the dictionary from data instead of using predefined bases has been shown to substantially improve signal reconstruction (Field and Olshausen 1996). We begin with a motivating example to illustrate the need of learning a dictionary from the data.

Example 1 Can a paralyzed man regain the motion of his hand? Ian Burkhart is a quadriplegic man who has become the first person to be implanted with technology that sends signals from the brain to muscles. This technological breakthrough is allowing him to regain some movement in his right arm and wrist. In 2014, scientists at Ohio State's Neurological Institute implanted a microchip into the 24-year-old Ian's motor cortex (see Bouton et al. 2016). Its goal is to bypass his damaged spinal cord so that with the help of a signal decoder and electrode-packed sleeve, he can control his right arm with his thoughts. The researchers need to decipher which brain signal is responsible for finger movement. In this process, they use wavelet decomposition, a technique equivalent to dictionary learning with a fixed basis. As more neural signals are acquired, an adaptive basis like one that dictionary learning provides will likely be more effective in this pursuit.

Dictionary learning. This last example in computational neuroscience illustrates the need of a powerful tool to characterize the collection of signals by sparse linear combinations of prototypical signals. Formally, consider n signals $y_1, y_2, \ldots, y_n$, each in $\mathbb{R}^d$, where $Y \in \mathbb{R}^{d\times n}$ is the matrix of *signals* (also "*samples*"). Additionally, assume that there exists $D \in \mathbb{R}^{d\times r}$, which is a *dictionary* of r prototypical signals, and $X \in \mathbb{R}^{r\times n}$, which is a *sparse matrix* of coefficients. In the parlance of dictionary learning, the columns of the dictionary D are the *atoms* $a_1, a_2, \ldots, a_r$. It is assumed that each signal is approximately equal to a linear combination of s atoms. For example, for y_i, there exist atoms $a_{i_1}, a_{i_2}, \ldots, a_{i_s}$ and coefficients c_{i_j}, $j = 1, 2, \ldots, s$, such that $y_i = c_{i_1}a_{i_1} + c_{i_2}a_{i_2} + \cdots + c_{i_s}a_{i_s}$. That is, formally, we assume that $Y = DX$. Dictionary learning attempts to recover a dictionary $\widehat{D}$ and matrix $\widehat{X}$ such that $Y \approx \widehat{D}\widehat{X}$. Throughout the remainder of the paper, we will drop the hats shown on $\widehat{D}$ and $\widehat{X}$ (any reference to the true dictionary and true sparse matrix will be clear by context). We say that a vector x is s-sparse if the vector has at most s non-zero entries, and we denote this by $\|x\|_0 \leq s$. Learning a dictionary from the signals is equivalent to the following optimization problem:

$$\min_{D,X} \sum_{i=1}^{n} ||y_i - Dx_i||_2^2 \quad \text{such that} \quad ||x_i||_0 \leq s,\ i = 1, \ldots, n,$$

where x_i is the ith column of the matrix X. In our formulation of the problem, both D and X are unknown.

If an oracle can supply us with the dictionary D, then finding a sparse representation of some signal $y \in \mathbb{R}^d$ amounts to solving the optimization problem

$$(P1) \qquad \min_{x} \quad \|x\|_1, \quad \text{such that } y = Dx.$$

This problem is commonly known as Basis Pursuit, or compressed sensing. The foundational papers by Donoho, Candes, Romberg, and Tao (Donoho 2006; Candes et al. 2006a, b) have firmly established that minimizing the l_1-norm of the vector x will ensure that the solution vector x is sparse. For the problem (P1), orthogonal matching pursuit (Cai and Wang 2011; Davenport and Wakin 2010; Cohen et al. 2017) is an efficient method to solve for the unknown sparse vector x. Other algorithms that are more sophisticated from the optimization viewpoint can also be used. (See, e.g. Blumensath and Davies 2009; Foucart 2012; Daubechies et al. 2010.)

Define the coherence of the dictionary D by

$$\mu_0 = \max_{j \neq k} \left\{ \left| \langle a_j, a_k \rangle \right| : 1 \leq j \leq r, 1 \leq k \leq r \right\}.$$

Suppose each column of the dictionary $D \in \mathbb{R}^{d \times r}$ is normalized to have length one. If the coherence μ_0 satisfies $(2s-1)\mu_0 < 1$, then every s-sparse vector $x \in \mathbb{R}^r$ is exactly recovered from the vector $y = Dx$ by solving the l_1-norm minimization problem. For the purpose of recovering dictionary atoms, we assume that our dictionary has small coherence. For example, the union of two orthogonal basis with small mutual coherence, such as spikes and sinusoids, belongs to this scenario. (See Donoho and Elad 2003 for a discussion and other examples).

However, in our problem, the challenge is that given the data matrix Y, both the dictionary D and the coefficients matrix X are unknown.A number of algorithms attempt to solve the dictionary learning problem. Most algorithms can be described as alternating minimization procedures. These algorithms begin by initializing the dictionary to a random matrix, and then alternating between solving for the dictionary D and the sparse matrix X. That means at each iteration, there are two steps. First, the matrix D is held fixed, while the best sparse matrix X is determined. Next, using the matrix X just computed, the dictionary D is updated. The method of optimal directions (MOD) (Engan et al. 1999) solves for the dictionary at each iteration, by the method of least squares, and computes the sparse matrix by a sparse coding algorithm such as orthogonal matching pursuit (OMP) (Cai and Wang 2011; Tropp 2004). A more sophisticated approach is the K-SVD algorithm (Aharon et al. 2006). This widely used algorithm replaces the least squares step of MOD by a more granular operation which decomposes error in the dictionary on a per-column basis.

The technique of alternating minimization involves computationally intensive operations on large matrices that can take hours or days to converge. A creative idea is introduced at the Conference on Learning Theory (COLT, 2014) in Spain (Agarwal et al. 2014, 2016; Arora et al. 2014). Agarwal, et al. present a fast algorithm for initializing the dictionary D using a clustering procedure based on SVD to extract initial atoms (Agarwal et al. 2014). This step recovers the dictionary with bounded error, and is followed by an alternating minimization procedure. The authors state that this is the first known exact recovery algorithm for the overcomplete ($r > d$) dictionary case. Importantly, they also empirically verify that under a common data generating process for Y, the initialization step is not sufficient for obtaining a good approximation of the true dictionary D.

This raises a natural question: Is there a data generating process under which it is possible to recover the dictionary atoms through only an initialization step? If so, then we can avoid the computationally expensive task of alternating minimization. We will show that under certain model assumptions, the answer is yes.

1.1 Main Contributions

We perform theoretical and experimental analysis of learning the prototypical atoms of a large collection of signals. This paper makes three main contributions:

(1) We prove that the atoms in the dictionary can be nearly recovered, provided that there are enough signals in the sample. This is the qualitative version of our main theorem. See Theorem 1 for the precise statement. As a rule of thumb, $O(r^2 \log r)$ signals would be sufficient.

(2) We provide empirical evidence for a data generating process and conditions under which a modified initialization algorithm similar to that of Agarwal et al. (2014) nearly recovers the atoms of the true dictionary D. This discovery is important because it can obviate the requirement of performing a subsequent alternating minimization step that ensures exact recovery. Removing this procedure can reduce the computational cost of dictionary learning significantly.

(3) The K-SVD algorithm is a widely used algorithm and is a useful benchmark. There is one principal difference between the K-SVD method and the algorithm under consideration in this article. Note that while the K-SVD method can find a dictionary D that nearly recovers the data Y, the optimization algorithm does not attempt to recover the true dictionary that generates the data. In contrast, we propose an algorithm that can nearly recover all the atoms in the true dictionary.

1.2 Model Assumptions

There is a trade-off between model assumptions and sample complexity. If a model is sufficiently general to encompass a wide range of scenarios, then the price to pay is that a large number of signals is required to learn the true dictionary from the sample data. Conversely, for a more specialized model, a smaller number of sample signals would suffice. We make the following model assumptions throughout the article. The data-generating process described in Sect. 3 can be thought of as a special case of the general model.

(M1) Each sample signal $y \in \mathbb{R}^d$ is generated as $y = Dx$, where D is the true dictionary and the coefficient vector x is s-sparse, i.e. at most s entries in the vector x are non-zero. All the signals are placed in the signal matrix Y, so that $Y = DX$, where the coefficient matrix X has r rows and each column is s-sparse. We assume that $r \geq d$.

(M2) Each column of the dictionary D is normalized to have length one, i.e. each atom satisfies $\|a_j\|_2 = 1$, for $1 \leq j \leq r$. Assume the coherence μ_0 of D satisfies $(2s - 1)\mu_0 < 1$.

(M3) The non-zero entries of each column of X are pairwise independent, conditioned on the support. The columns of X are pairwise independent.

(M4) The entries in each column vector x_k has the following properties: (i) each entry $x_k(j)$ is drawn from the set $[-\beta, -\alpha] \cup [\alpha, \beta]$, where $\alpha, \beta > 0$, for $1 \leq j \leq r$ and $1 \leq k \leq n$, (ii) $E[x_k(j)] = 0$.

(M5) For the successful recovery of the dictionary D from the sample signals Y, the signals must be sufficiently sparse. We impose the following condition: $s \leq O(r^{1/3})$.

(M6) The support of each column vector in X is chosen uniformly and independently from subsets of size s.

1.3 Related Work

Our work is motivated by the clustering approach to dictionary initialization by Agarwal et al. (2014). Independently and concurrently, Arora et al. (2014) introduce a slightly different clustering procedure, under different model assumptions. A sophisticated approach that avoids alternating minimization by the use of tensor decomposition is introduced by Barak et al. (2015). There is a trade-off between sparsity of the signals and sample complexity in all these works, including ours. Either the signals need to be very sparse, i.e. sparser than $s \leq O(r^{1/2})$, or the number of signals required is at least $O(r^c)$, with $c \geq 3$. Methods that achieve equality in the sparsity bound above generally require technical assumptions that can be difficult to check. In contrast, we have avoided weaker assumptions that are more general, but offer less intuitive appeal. Our own requirement that $s \leq O(r^{1/3})$ sacrifices generality for clarity, but remains computationally simple to check and comparable with existing requirements.

Our requirement that $n \geq O(r^2 \log r)$ may seem conservative. However, in numerical experiments, when we pick a specific data generating process, a specific true dictionary with some specific values of r, then we find that number of samples needed is far less than $r^2 log(r)$. In groundbreaking work, Spielman et al. (2012) are the first ones who give an algorithm that can provably recover a dictionary that is a basis, with high probability. They prove that the number of samples needed are $n = O(r \log r)$. As noted by the authors, their analysis does not extend to an overcomplete dictionary. See also the discussion in Gribonval et al. (2015); Gribonval and Schnass (2010) and Schnass (2014) regarding the number of samples needed to identify a dictionary.

Remark 1 will explain why we cannot expect to recover the dictionary atoms with less than $O(r \log r)$ samples.

2 Initializing Dictionaries for Fast Optimization

The core insight of the initialization algorithm InitDictionaryLearn of Agarwal et al. is that the atoms extracted from the data should be limited to those that represent clusters of signals. The algorithm tests pairs of signals to find those pairs that share an atom, then finds signals that are correlated with the pair, which forms a cluster of signals. If the cluster is "good" (a decision determined by Agarwal's UniqueIntersection algorithm), then InitDictionaryLearn extracts an atom in a process similar to PCA, using information from every entry of the signals in the cluster. We modify the algorithms InitDictionaryLearn and UniqueIntersection presented in Agarwal et al. (2014), and name our modifications P1 and P2, respectively. The algorithm P1 is outlined in Algorithm 2, and P2 is outlined in Algorithm 1. Our modifications follow.

We evaluate the algorithm on the result of a data generating process wherein the original dictionary D is a low coherence dictionary, and the elements of the columns of the true sparse matrix X are integers which have restrictions on their magnitude and distribution across entries of each x_i. A description of this data generating process is in Sect. 3.

We formulate a new correlation threshold τ specifically for our data generating process based on the assumption that the columns of our sparse matrix take on certain values in the worst case. Our correlation threshold's calculation implies additional restrictions for the data generating process. A description is given in Sect. 5.

We also formulate a different threshold ϵ_1 for use in P2 for the average correlation between signals in a cluster detected by P1. This threshold is used to filter out clusters that don't contain atoms sharing the same signal, and its formulation is based on intuition given by probabilistic estimations under some assumptions described in Sect. 4.

Algorithm 1 Verifying candidate atom cluster S contains a true atom. Note: p_0 and p_1 refer to the first and second entries of the tuple p.

```
procedure P2
    Input cluster of signals $S$, number of atoms $r$, sparsity $s$, correlation threshold $\tau$.
    Initialize $P \leftarrow$ exclusive consecutive pairs in $S$, $c \leftarrow 0$, $M \leftarrow |P|$, $\epsilon_1 \leftarrow 1 - \frac{19s^3}{r}$.
    for $p \in P$ do
        $i, j \leftarrow p_0, p_1$
        if $|\langle y_i, y_j \rangle| > \tau$ then
            $c \leftarrow c + 1$
        end if
    end for
    return $c/M \geq \epsilon_1$
end procedure
```

Algorithm 2 Clustering signals in A for extracting r true atoms, assuming each signal is represented by an s-sparse sum of the atoms. The EIGTOP function returns the top eigenvector of a matrix. "$|\langle\cdot\rangle|$" is the unsigned inner product.

```
procedure P1
    Input signals $Y$, number of atoms $r$, sparsity $s$, correlation threshold $\tau$,
     minimum separation between recovered atoms in norm difference $\epsilon_A$,
     and coherence of true dictionary $\mu$.
    Initialize $A \leftarrow \emptyset$, $L \leftarrow \emptyset$.
    while $|A| \leq r$ do
        Randomly pick a pair of signals
          $(y_p, y_q)$ from $2^Y \setminus L$.
        $L \leftarrow L \cup \{(p, q)\}$.
        if $|\langle y_p, y_q \rangle| > \tau$ then
            $S \leftarrow$ the set of all signals $z$ such that $|\langle y_p, y_z \rangle| > \tau$ and $|\langle y_z, y_q \rangle| > \tau$
            if $|S|$ mod 2 is 0 then
                $z_{min} \leftarrow$ signal in $S$ with smallest average inner product with $p$ and $q$
                $S \leftarrow S \setminus \{z_{min}\}$
            end if
            if $|S| \geq 2$ and
                P2($S, r, s, \tau$) then
                $B \leftarrow \sum_{i=1}^{|S|} z_i z_i^T$.
                $u \leftarrow$ EIGTOP($B$).
                if $\min_{a \in A} |a - u| > \epsilon_A$ then
                    $A \leftarrow A \cup \{u\}$.
                end if
            end if
        end if
    end while
    return $A$.
end procedure
```

3 Data Generating Process

Agarwal et al. (2014) test a data generating process where entries of D are drawn from $\mathcal{N}(0, 1)$, the support of each column vector in X is chosen uniformly and independently from subsets of size s, and the non-zero values of each X column vector are chosen uniformly and independently from $[-2, -1] \cup [1, 2]$. Although these features of their process are not necessary for their theoretical work, we construct a similar data generating process in order to match this existing literature.

Our data generating process is a choice of a true dictionary D and s-sparse matrix X. The signals generated are defined by $Y = DX$. This process is inspired by problems in classical signal processing–recovery of signals created by low coherence dictionaries. Consequently, a low coherence matrix is chosen as the dictionary. This is our first main modification of the data generating process of Agarwal et al. (2014). Like Agarwal et al., we consider the case where the signal matrix $Y \in \mathbb{R}^{d \times n}$ has $d < n$. Our second main modification is as follows: we choose three integers α, β, and γ, with γ positive. Like Agarwal et. al., we choose the locations of the non-zero entries uniformly and independently from the subsets of size s. We set one non-zero entry of each column to be β, and the rest are drawn uniformly independently from $\{-\alpha\} \cup \{\alpha\}$. This implies that the non-zero entries of each column are in the set $\{-\alpha, \alpha, \beta\}$. We further insist that no more than γ of the β-valued elements exist in the same entry of any subset of column vectors of X. This condition ensures that β-valued elements are not clustered together in dimension. Additionally, the use of γ makes the performance of our experiments more easily verifiable, and only slightly differs from assumptions (M3) and (M6). Finally, this data generating process implies that any procedure clustering these vectors by the correlation function demands an additional condition; that

$$\beta^2 - 2\alpha\beta > 2\alpha^2(s-2).$$

This condition is derived in Sect. 5.

Our restriction to integer-valued elements, restriction on the relative sizes and dimensional distributions of the elements, and use of a low coherence dictionary are the major differences between our data generating process and that of Agarwal et al. We accept these restrictions because of their relationship to realistic problems, e.g. where a signal may have one predominant atom that has been corrupted by "noise" from other atoms, and where each atom is distinct (i.e. D has low coherence). However, this is only one realization of our general data generating process described by assumptions (M1) to (M6), and our general process offers flexibility to encompass other assumptions as well.

Several extensions to our data generating process can be made from relaxing existing assumptions as well. It is not complicated to remove the γ parameter entirely. As each column of $\mathbf{X}$, β is chosen uniformly and independently from subsets of size s, γ can instead be calculated by examining the placement of each β, instead of being assumed. In this formulation of the data generating process, γ becomes a random variable that we *do not control*, and which is distributed according to a multinomial distribution with mean n/d and variance $(nd-n)/d^2$. With this extension, our data generating process more closely matches that of Agarwal et al. Notably, in all of our experiments where γ is fixed, we let γ equal n/d as well.

Our data generating process can also be extended to allow the selection of values other than those in the set $\{\alpha, \beta\}$ for the non-zero entries of $\mathbf{X}$. It is simplest to relax the condition on α. As long as the threshold condition derived in Sect. 5 is not violated, α can be replaced with an interval of values smaller than α and greater than zero. Consequently, we can select values from $\beta \cup [\epsilon, \alpha]$, where $\epsilon > 0$. The extension of β from an integer to an interval is not trivial, and would complicate our analysis. The spiked signal model considered in this work is simplified by having one large coefficient corresponding to one atom. We suspect that it is possible to extend β to an interval of values larger than β, but such an extension would require additional analysis, and we leave this to future work.

4 Probabilistic Bounds

In P1, our goal is to find clusters of signals that may share the same atom; that is, signals y_p and y_q "share an atom" if X_{mp} and X_{mq} are both non-zero for some row m of X. These potential clusters are constructed by selecting pairs of signals (y_p, y_q) that have large inner product (lines 10–11), and then finding other signals that have large inner product with each signal in the pair (lines 14–16). Once we identify correlated clusters of signals in P1, we extract an atom through the process in lines 25–29 (Agarwal et al. 2014).

Under our proposed data process and correlation threshold, lines 10–16 select clusters with signals that all share the same atom with coefficient β. However, under other data processes, there is no assurance that this will happen. Consequently, this process may select "bad" clusters which contain signals that might not (a) share a single unique atom and (b) have non-negligible contributions from other atoms (in our data process, this implies that the coefficient on these atoms is greater than α).

To gain insight into the probability that each cluster identified by $P1$ is "good", we analyze the probability that any pair of signals in a cluster shares the same unique atom, but not any other atoms. We introduce the following scenario and events to formalize this problem: pick two signals from the data, y_p and y_q, and consider two arbitrary signals y_i and y_j. Define the following events:

- $SU(y_p, y_q)$: The sums that represent y_p and y_q share exactly one unique atom.
- $SU(y_i, y_j)$: The sums that represent y_i and y_j share exactly one unique atom.
- E_1: y_i shares *exactly* one atom with y_p, and y_i shares *exactly* one atom with y_q. Also, y_j shares *exactly* one atom with y_p, and y_j shares *exactly* one atom with y_q.
- F_1: y_i shares *at least* one atom with y_p, and y_i shares *at least* one atom with y_q. Also, y_j shares *at least* one atom with y_p, and y_j shares *at least* one atom with y_q.

$SU(y_p, y_q)$ corresponds to picking the initial pair of correlated signals, because if $|\langle y_p, y_q\rangle| > \tau$, then y_p and y_q share at least one atom.

To make the analysis tractable, we assume that this shared atom is the only atom y_p and y_q share, even though P2 may select pairs that share more than one unique atom. Additionally, we know that if, for some signal y_z, $|\langle y_p, y_z\rangle| > \tau$ and $|\langle y_q, y_z\rangle| > \tau$, then y_z shares at least one atom with each of y_p and y_q. F_1 defines this event for some pair of signals (y_i, y_j). We are interested in the following probability: given that $SU(y_p, y_q)$ and F_1 have occurred, what is the probability that the events E_1 and $SU(y_i, y_j)$ will occur? We are interested in analyzing

$$P[SU(y_i, y_j) \cap E_1 | F_1 \cap SU(y_p, y_q)], \tag{1}$$

In other words, if we have a pair (y_p, y_q) which share a unique atom, and another candidate pair (y_i, y_j), each of which shares at least one atom with y_p and y_q, what is the probability that (y_i, y_j) share the same unique atom with each other (this is $SU(y_i, y_j)$) that they uniquely share with y_p and y_q (this is E_1)?

Knowing a lower bound on (1) allows us to select only those clusters in which enough candidate pairs of signals from the cluster are correlated with each other to, on average, share a unique atom. In P2, we again estimate correlation by checking to see if two signals have inner product with a magnitude greater than τ. We count all signals that pass this criterion, and use this count as an empirical estimator of (1). If this empirical estimation of (1) is above the lower bound on (1) required for the signals to share a unique atom, P2 returns TRUE, and we continue on to the rest of P1 (that is, lines 25–29 which extract an atom).

It is notable that this procedure does not depend on the data process we establish above, and that τ may be calculated differently for a separate data process without affecting the above calculations.

5 Correlation Threshold

We would not like to extract atoms from a cluster formed by P1 if the signals do not all share an atom. We calculate a correlation threshold in order to detect and reject clusters of signals fitting this description. We derive the correlation threshold based on the worst-case inner product of two vectors which do not share the same atom.

If two vectors in Y do not share the same atom, the β-valued element is not contained in the same entry. Consequently, we know that the inner product will be at most

$$\tau = 2\alpha\beta + \alpha^2(s-2)$$

in magnitude, as each β-valued entry may, by chance, be multiplied by a signal with α in the same entry with the same sign, leaving $s-2$ potential α-valued entries with the same sign. In these calculations, we ignore the elements of D, as each element in D is bounded in magnitude by 1, and therefore the product of any of these elements will not affect this upper bound on the inner product between two vectors in Y that do not share atoms with coefficient β.

Importantly, we must make sure that this threshold does not bar clusters comprised entirely of signals that share the same atom from being selected. In this case, without loss of generality, the worst-case result is that the β-valued entry is positive, and that the α-valued entries are all of opposite sign; therefore these entries decrease the magnitude of the inner product of two signals which share an atom. Thus we gain the restriction that

$$\beta^2 - \alpha^2(s-2) > \tau_1 \Leftrightarrow \beta^2 - 2\alpha\beta > 2\alpha^2(s-2).$$

6 Proof of the Main Theorem

The presentation of our theorem and its proof would be cleaner if we adopt the following convention. We say that an event E occurs with an overwhelmingly high probability if $P(E) \geq 1 - O(1/r^2)$, i.e. it occurs, except with probability at most $O(1/r^2)$. Fix a constant $\delta \leq 0.02$. We say that a dictionary atom a_j is nearly recovered if our dictionary initialization procedure yields a vector $\widehat{a}_j$ such that $\|\widehat{a}_j - \epsilon a_j\| \leq \delta$, where $\epsilon = 1$ or -1 to account for the sign ambiguity. In the following work, the letters c_1 and k_1 denote small constants and we do not keep track of their precise values.

Assume the model assumptions (M1) to (M6). Our theoretical guarantee for the successful recovery of the true dictionary can be stated as:

Theorem 1 *By using the dictionary initialization procedure described in Sect.* 2 (*Algorithm 1 and* 2)*, under our model assumptions, the atoms in the true dictionary can be nearly recovered, with an overwhelmingly high probability, provided that the number of samples* $n \geq O(r^2 \log r)$.

Remark 1 Our requirement that $n \geq O(r^2 \log r)$ may seem conservative. Let us explain why we cannot expect to recover the dictionary atoms with less than $O(r \log r)$ samples.

Consider the following generalization to the coupon collector problem. There are n urns and coupons are placed at random in these urns. Each coupon may be placed in any of the urns with the same probability. The coupons are placed into the urns, one at a time, and the choices of the urns for the different coupons are independent. We continue this process until there are at least m coupons in each urn. Let $T_m(n)$ be the expected number of coupons that are needed. Newman and Shepp (1960) show that if we fix the value of m, then as the value of n gets large, $T_m(n)$ grows asymptotically as $O(n \log n + (m-1)\log(\log n))$. If we want each atom to be shared by at least 5 signals, then by assumption (M6) and the reasoning of the generalized coupon-collector problem, we expect that we need at least $r \log r + 4 \log(\log r)$ samples.

Plan of the Proof Before giving the proof of the theorem, let us give an outline of the plan.

The relation $Y = DX$ can be captured by a bipartite graph B_{data}, consisting of Red and Blue vertices. Each signal is a vertex in the set of Red vertices. Each atom in the dictionary is a vertex in the set of Blue vertices. An edge connects a Red vertex y_k to a Blue vertex a_j if and only if the signal y_k has a non-zero coefficient for the atom a_j. For example, if $y_2 = 3a_6 + 4a_7 + 5a_8$, then the red vertex y_2 is connected to the blue vertices a_6, a_7, a_8. If two signals y_1 and y_2 share an atom a_6 in common, then this Blue vertex a_6 is a common neighbour of these two Red vertices. In this way, we see that the edges in the bipartite graph B_{data} encodes the sparsity pattern of the coefficient matrix X. We write $N_B(y_k) = \{a_u\}$ if an atom a_u is a neighbour of vertex y_k, i.e. if there is an edge connecting y_k to a_u. Consider a particular atom. The set of all signals that share this atom as a common neighbour forms a cluster for that atom. The plan is to show that we can recover the atom if there are enough signals in the cluster.

Definition 1 Fix a pair of atoms y_p and y_q such that the intersection of each signal's set of atoms contains only one unique atom. That is, let $N_B(y_p) \cap N_B(y_q) = \{a_z\}$, for some atom a_z at the zth column index of D. Then for each candidate signal y in the signal matrix Y, define the following sets:

$$S_A = \{y : |\langle y, y_p\rangle| > \tau\} \qquad S_B = \{y : |\langle y, y_q\rangle| > \tau\}$$

Define $S_{AB} = S_A \cap S_B$.
Then G is the collection of good signals for that atom, defined by

$$G = \{y \in S_{AB} : N_B(y) \cap N_B(y_p) = \{a_z\} \text{ and } N_B(y) \cap N_B(y_q) = \{a_z\}\}.$$
$$H = \{y \in S_{AB} : y \notin G\}.$$

The intuition of our proof is as follows. Recall that for a cluster of atoms S_{AB} identified by Algorithm 2, we let $B = \sum_{y_k \in S_{AB}} y_k y_k^T$. Then by the above definitions, $B = \sum_{y_k \in G} y_k y_k^T + \sum_{y_k \in H} y_k y_k^T$. Our goal is to recover the atom a_z from y_k, while avoiding the noise added by b_k. Therefore, in the matrix B, we wish to show that the uncorrelated noise from the b_k vectors essentially cancel out, leaving the atom that has coefficient β (denoted as a_z) to be extracted through the EIGTOP procedure. Concretely, let us define $v_1 = \text{EIGTOP}(B)$. Then we wish to show that $||v_1 - \epsilon a_z||_2 \leq \delta$.

Remark 2 The intuition described above provides a useful roadmap for the proof. Turning this intuition into a proof requires some care. One may initially suspect that for a given matrix A_1, if A_2 is a small perturbation of A_1 (i.e. A_2 is just A_1 with a small amount of noise added), then the corresponding top eigenvectors of the two matrices will be close to each other. We include the following example to illustrate what can go wrong.

Example 2 Consider two 3×3 diagonal matrices, $A_1 = \text{diag}(1 + \epsilon, 1, 1)$, and $A_2 = \text{diag}(1, 1 + \epsilon, 1)$. Then A_2 is a small perturbation of A_1. Indeed, $\|A_1 - A_2\| \leq 2\epsilon$. However, the top eigenvector for A_1 and A_2 are e_1 and e_2, respectively. Here, e_1 and e_2 are the first 2 columns of the 3×3 identity matrix. These eigenvectors are orthogonal to each other.

This example serves as a cautionary tale. Thus the proof entails some careful analysis. For a fixed matrix, we cannot conclude the top eigenvector of the matrix will be changed by a small amount, when the matrix is subject to a small perturbation. What we want is an analogous statement about a large random matrix undergoing a small random perturbation.

We will use the following result on random matrices (Vershynin 2012). Hidden in the background is a deep result on the convexity of operator algebras (see, e.g. Hansen and Pedersen 2003).

Theorem 2 *Let A be an N by n matrix whose rows A_i are independent random vectors in $\mathbb{R}^n$ with the common second moment matrix $\Sigma = E[A_i \otimes A_i]$. Suppose $\|A_i\|_2 \leq \sqrt{K}$ almost surely for all i. Then, for each $t > 0$, with probability at least* $1 - n\exp(-ct^2)$,

$$\left\| \frac{1}{N} A^* A - \Sigma \right\| \leq \max(\|\Sigma\|^{1/2}\delta, \delta^2), \quad \textit{where } \delta = t\sqrt{\frac{K}{N}}.$$

Here, $c > 0$ is an absolute constant. In particular,

$$\|A\| \leq \|\Sigma\|^{1/2}\sqrt{N} + t\sqrt{K}.$$

Definition 2 For each vector in $y_k \in G$, we define $b_k = y_k - x_k(z)a_z$, where by $x_k(z)$ we denote the zth entry of the vector at the kth column of the matrix X. A special role will be played by the $d \times |G|$ matrix of all vectors in G.

To prepare for the proof of Theorem 1, we proceed with a series of intermediate steps. We begin by bounding the norms of b_k and y_k with a small observation.

Remark 3 Denote the mutual coherence of the matrix D by μ_0. If $\mu_0 \leq \frac{1}{2s-1}$, then $||b_k||_2 \leq 2 \cdot \beta\sqrt{s}$ and $||y_k||_2 \leq \beta\sqrt{2s}$.

Proof From the definition of the inner product,

$$\begin{aligned}
||y_k||_2^2 = \langle y_k, y_k \rangle = \left\langle \sum_{a_j \in N_B(y_k)} x_k(j)a_j, \sum_{a_l \in N_B(y_k)} x_k(l)a_l, \right\rangle \\
&= \sum_{a_j, a_l \in N_B(y_k)} x_k(j)x_k(l)\langle a_j, a_l \rangle \\
&\leq \sum_{a_j, a_l \in N_B(y_k)} |\langle a_j, a_l \rangle| \cdot |x_k(j)x_k(l)| \\
&= \sum_{a_j \in N_B(y_k)} x_k(j)^2 ||a_j||_2^2 + \sum_{a_j \neq a_l \in N_B(y_k)} |x_k(l)x_k(j)| \cdot |\langle a_j, a_l \rangle| \\
&\leq \beta^2 s + \beta^2 s^2 \mu_0 \leq \frac{3s}{2}\beta^2.
\end{aligned}$$

Because $||b_k||_2 \leq ||y_k||_2 + |x_k(z)| \cdot ||a_z||_2$, it follows that $||b_k||_2 \leq ||y_k||_2 + \beta$. □

Our next task is to control the size of $\| \sum_{b_k \in G} b_k b_k^T \|$.

Proposition 1 *If* $\|D^T\|_{op} \leq \sigma_1$, *then*

$$\left\| E[b_i b_i^T] \right\| \leq \beta^2 \sigma_1^2 \frac{s}{r}.$$

Proof From Remark 3, $||b_i||_2 \leq 2\beta\sqrt{s}$. Fix some $h \in \mathbb{R}^d$ with $||h||_2 = 1$. Note that

$$h^T E[b_i b_i^T]h = E[h^T b_i b_i^T h] = E[(h^T b_i)^2].$$

Now let $v = D^T h$, and note that $v \in \mathbb{R}^r$, where r is the number of atoms. Define the vector $\hat{x}_i$ to be equal to x_i, but with $\hat{x}_i(z) = 0$. Then

$$\begin{aligned}
E[(h^T b_i)^2] = E[(v^T \hat{x}_i)^2] \leq E\left[\sum_{j=1}^{r} v(j)^2 \hat{x}_i(j)^2\right] = E\left[\sum_{j \neq k}^{r} v(j)v(k)\hat{x}_i(j)\hat{x}_i(k)\right] \\
&\leq \sum_{j=1}^{r} v(j)^2 E[\hat{x}_i(j)^2] + \sum_{j \neq k}^{r} |v(j)v(k)| \cdot |E[\hat{x}_i(j)\hat{x}_i(k)]| \\
&\leq \sum_{j=1}^{r} v(j)^2 \beta^2 \frac{s}{r}.
\end{aligned}$$

The last inequality is valid because for each entry in $\hat{x}_i$, only s/r of them are nonzero, and all are less than β in absolute value. Note that in the second to last inequality, the summation over $j \neq k$ is zero because the random variables $\hat{x}_i$ are independent. Thus

$$h^T E[b_i b_i^T]h \leq \beta^2 \frac{s}{r} ||v||_2^2,$$

where

$$||v||_2 = ||D^T h||_2 \leq ||D^T||_{op}||h||_2 = ||D^T||_{op} \leq \sigma_1^2$$

and the conclusion follows. □

Given Proposition 1, we have that $||E[b_i b_i^T]|| \leq \beta^2(s/r)\sigma_1^2$. This essentially bounds the contribution of the noisy parts of each signal to the matrix supplied to the EIGTOP procedure. Now, we return our attention to the matrix B.

Definition 3 We define the matrices

$$B_1 := \sum_{y_k \in G} y_k y_k^T \qquad \text{and} \qquad B_2 := \sum_{y_k \in H} y_k y_k^T$$

so that $B = B_1 + B_2$. Note that B_1 is the $d \times |G|$ matrix of all vectors in G.

The next proposition directly follows from Theorem 2 by paraphrasing the theorem.

Proposition 2 *Let B_3 be a matrix of size $d \times |G|$ defined by*

$$B_3 := \sum_{b_i \in G} b_i b_i^T$$

and suppose each vector b_i satisfies

$$||b_i||_2^2 \leq (2\beta\sqrt{s})^2 := u.$$

Let c_1 be some positive constant, and define $t := c_1 \cdot \sqrt{|G|}$. Define

$$||\Sigma|| := ||E[b_i b_i^T]|| \leq \sigma_1^2 \beta^2 \frac{s}{r}.$$

Then the matrix B_3 satisfies

$$||B_3||_{op} \leq ||\Sigma||^{1/2}\sqrt{|G|} + t\sqrt{u}$$

with probability at least $1 - d \cdot \exp(-ct^2)$, where $c > 0$ is an absolute constant.

We have the following immediate consequence:

Proposition 3 *With probability at least* $1 - d \cdot \exp\left(-cc_1^2|G|\right)$,

$$\left\| \sum_{b_k \in G} b_k b_k^T \right\|_{op} \leq 2\beta^2 s|G| \left(\frac{\sigma_1^2}{r} + 4c_1^2 \right).$$

Proof By applying Proposition 2, we see that

$$||B_3||_{op} \leq \sigma_1 \sqrt{\frac{s}{r}} \beta \sqrt{|G|} + 2\beta\sqrt{s} \cdot c_1 \sqrt{|G|} \leq \beta\sqrt{s}\sqrt{|G|} \left(\frac{\sigma_1}{\sqrt{r}} + 2c_1 \right)$$

with probability at least $1-d\cdot\exp\left(-cc_1^2|G|\right)$. From the inequality $(a+b)^2 \leq 2(a^2+b^2)$ for any a and b, we have

$$||B_3 B_3^T||_{op} \leq ||B_3||_{op}^2 \leq \left[\beta\sqrt{s}\sqrt{|G|} \left(\frac{\sigma_1}{\sqrt{r}} + 2c_1 \right) \right]^2 \leq 2\beta^2 s|G| \left(\frac{\sigma_1^2}{r} + 4c_1^2 \right).$$

□

Recall that $y_k = x_k(z)a_z + b_k$. Expanding B_1, we see that

$$B_1 = \sum_{y_k \in G} x_k(z)^2 a_z a_z^T + \sum_{y_k \in G} x_k(z)(a_z b_k^T + b_k a_z^T) + \sum_{y_k \in G} b_k b_k^T.$$

To analyze how each of the last two matrices in the sum above contribute to B_1, and how B_2 contributes to B, we bound the relevant operator norms.

Proposition 4 *The following bounds exist with probability at least* $1-d\cdot\exp\left(-cc_1^2|G|\right)$, *where* $c > 0$ *is an absolute constant.*

$$\left\| \sum_{y_k \in G} x_k(z)^2 a_z b_k^T \right\|_{op} \leq \beta^2|G|s \left(\frac{\sigma_1}{\sqrt{rs}} + \frac{2c_1}{\sqrt{s}} \right), \tag{2}$$

$$\left\| \sum_{y_k \in G} b_k b_k^T \right\|_{op} \leq 2\beta^2|G|s \left(\frac{\sigma_1^2}{r} + 4c_1^2 \right), \tag{3}$$

$$\|B_2\|_{op} \leq 2|H|\beta^2 s \tag{4}$$

Proof We start by bounding Eq. (2). As there are $\sqrt{|G|}$ $x_k(z)$ vectors in the first norm, and for each, $|x_k(z)| \leq \beta$, from the definition of the operator norm, we see that

$$\left\| \sum_{y_k \in G} x_k(z)^2 a_z b_k^T \right\|_{op} \leq ||a_z||_2 ||B_3||_{op} \cdot \beta\sqrt{|G|}.$$

Then by Proposition 2, and because $||a_z||_2 = 1$,

$$\begin{aligned} ||a_z||_2||B_3||_{op} \cdot \beta\sqrt{|G|} &\leq \left(\beta\sqrt{s}\sqrt{|G|}\sqrt{\sigma_1^2/r + 2c_1}\right)\beta\sqrt{|G|}(1) \\ &= \beta^2|G|s\left(\frac{\sigma_1}{\sqrt{rs}} + \frac{2c_1}{\sqrt{s}}\right). \end{aligned}$$

For Eq. (3), this follows directly from Proposition 3,

$$\left\|\sum_{y_k \in G} b_k b_k^T\right\|_{op} = ||B_3 B_3^T||_{op} \leq 2\beta^2|G|s\left(\frac{\sigma_1^2}{r} + 4c_1^2\right).$$

For Eq. (4), because each of the s entries of x_k is less than or equal to β, we see that by Remark 3,

$$||B_2||_{op} = \left\|\sum_{y_k \in H} y_k y_k^T\right\|_{op} \leq |H| \cdot ||y_k||_2^2 \leq |H|2s\beta^2.$$

□

Now that we have bounded the matrices supplying "noise" to B, we can prove that a_z is close to the top eigenvector of B, thus showing that our clustering algorithm extracts the desired atom.

Proposition 5 *Let v_1 be the first eigenvector of the matrix B. Define the variance*

$$\mathbb{V} = \frac{1}{|G|}\sum_{y_k \in G} x_k(z)^2.$$

Then, with probability at least $1 - d \cdot \exp\left(-cc_1^2|G|\right)$, we have

$$||v_1 - \epsilon a_z||_2^2 \leq \frac{4\beta^2 s k_1}{\alpha^2},$$

where $\epsilon \in \{-1, 1\}$ and k_1 is a small constant.

Proof Let $\cos\theta = \langle v_1, \epsilon a_z\rangle$. To bound the difference between the atom and the top eigenvector of the matrix, we begin with

$$||v_1 - \epsilon a_z||_2^2 = \langle v_1 - \epsilon a_z, v_1 - \epsilon a_z\rangle = ||v_1||_2^2 + ||a_z||_2^2 - 2\langle v_1, \epsilon a_z\rangle.$$

By hypothesis, both vectors are normalized to length one, so

$$||v_1 - \epsilon a_z||_2^2 = 2 - 2\cos\theta. \tag{5}$$

Next, we seek an upper bound and a lower bound on the size of the matrix B. The operator norm of B coincides with the largest eigenvalue of B, since B is self-adjoint, which means that $\|B_1\|_{op} = ||v_1^T B v_1||_2$. For an upper bound, we compute

$$\begin{aligned}\left\|v_1^T B v_1\right\|_2 &= \left\|v_1^T (B_1 + B_2) v_1\right\|_2 \\ &\leq \cos^2\theta \cdot \mathbb{V}|G| + 2\left\|\sum_{y_k \in G} x_k(z) a_z b_k^T\right\|_{op} + 2\left\|\sum_{y_k \in G} b_k b_k^T\right\|_{op} + ||B_2||_{op} \\ &\leq \cos^2\theta \cdot \mathbb{V}|G| + 2\left[\beta^2 |G| s\left(\frac{\sigma_1}{\sqrt{rs}} + \frac{2c_1}{\sqrt{s}}\right)\right] \\ &\quad + 2\beta^2|G|s\left(\frac{\sigma_1^2}{r} + 4c_1^2\right) + \left(2s\beta^2|H|\right) \leq \beta^2 \cdot |G|\left(\frac{\mathbb{V}\cos^2\theta}{\beta^2} + k_1 s\right).\end{aligned}$$

To obtain a lower bound, we essentially replace the triangle inequality with the reverse triangle inequality in the above calculation.

We have $||B|| \geq ||B_1|| - ||B_2||$. Next, we compute

$$\begin{aligned}||B||_{op} &\geq \mathbb{V}|G| \cdot ||a_z||_2^2 - 2\left\|\sum_{y_k \in G} x_k(z) a_z b_k^T\right\|_{op} - 2\left\|\sum_{y_k \in G} b_k b_k^T\right\|_{op} - ||B_2|| \\ &\geq \mathbb{V}|G| - 2\left[\beta^2 \cdot |G| \cdot s\left(\frac{\sigma_1}{\sqrt{rs}} + \frac{2c_1}{\sqrt{s}}\right)\right] \\ &\quad - 2\beta^2 \cdot |G| \cdot s\left(\frac{\sigma_1^2}{r} + 4c_1^2\right) - \left(|H| \cdot 2s\beta^2\right) \geq \beta^2|G|\left(\frac{\mathbb{V}}{\beta^2} - k_1 s\right).\end{aligned}$$

We now have the sandwich inequalities,

$$\beta^2 \cdot |G|\left[\frac{\mathbb{V}}{\beta^2} - k_1 s\right] \leq ||v_1^T B v_1||_2 \leq \beta^2 \cdot |G|\left[\frac{\mathbb{V}\cos^2\theta}{\beta^2} + k_1 s\right],$$

and so,

$$\frac{\mathbb{V}}{\beta^2} \leq \frac{\mathbb{V}}{\beta^2}\cos^2\theta + 2k_1 s,$$

which is equivalent to the lower bound,

$$\cos^2\theta \geq 1 - \frac{2\beta^2}{\mathbb{V}} s k_1. \tag{6}$$

Recall that the variance $\mathbb{V}$ is given by

$$\mathbb{V} = \frac{1}{|G|} \sum_{y_k \in G} x_k(z)^2$$

and for each term in the summation, $x_k(z) \geq \alpha$ by our model assumption. Therefore,

$$\frac{-1}{\mathbb{V}} \geq \frac{-1}{\alpha^2}. \tag{7}$$

Comparing Eqs. (6) and (7), we conclude that

$$\cos^2\theta \geq 1 - 2\frac{\beta^2}{\alpha^2} s k_1. \tag{8}$$

From Eq. (5), and for $0 \leq \cos\theta \leq 1$, we have $\|v_1 - \epsilon a_z\|_2^2 \leq 2(1 - \cos^2\theta)$. Finally, combining this with Eq. (8), we arrive at the desirable conclusion,

$$||v_1 - \epsilon a_z||_2^2 \leq 2\left(2\frac{\beta^2}{\alpha^2} s k_1\right)$$

with probability $1 - d \cdot \exp(-cc_1^2|G|)$.
This completes the proof. □

We have established that each dictionary atom can be recovered with overwhelmingly high probability, provided that the size of the set G is sufficiently large. The next task is to prove that with high probability, G contains at least $\frac{3ns}{8r}$ signals. We will use the following classical inequality for a sum of random variables.

Lemma 1 (Hoeffding inequality) *Let $X_1, X_2, \ldots, X_n$ be independent random variables. Assume that each X_j is bounded, i.e. $a_j \leq X_j \leq b_j$ for $1 \leq j \leq n$. Then, for the sum of these variables $S = X_1 + \cdots X_n$, we have*

$$P\left(S - E(S) \leq -t\right) \leq \exp\left(-\frac{2t^2}{\sum_{j=1}^n (b_j - a_j)^2}\right)$$

which is valid for all $t > 0$. $E(S)$ is the expected value of S. In particular, for the choice of $t = E(S)/2$, and when $a_j = a$ and $b_j = b$ for each j, we have

$$P\left(S \leq \frac{E(S)}{2}\right) \leq \exp\left(-\frac{[E(S)]^2}{2n(b-a)^2}\right).$$

Proposition 6 *Fix a pair of signals y_p and y_q such that they share exactly one atom in common. That is, $N_B(y_p) \cap N_B(y_q) = \{a_z\}$, for some atom a_z from one column of* *matrix D. With the same definition of G, the "good" set of signals for that particular atom, we have that*

$$P\left(|G| \geq \frac{3ns}{8r}\right) \geq 1 - \exp\left(-\frac{9ns^2}{32r^2}\right). \tag{9}$$

Proof Let y be any signal from the matrix Y. Let E be the event that y belongs to set G. If y belongs to G, then y must share exactly one atom in common with both y_p and y_q. More precisely, $N_B(y) \cap N_B(y_p) = \{a_z\}$ and $N_B(y) \cap N_B(y_q)| = \{a_z\}$.
Since $|N_B(y_p) \cap N_B(y_q)| = 1$, we know that $|N_B(y_p) \cup N_B(y_q)| = 2s - 1$.
For the event E to occur, we must pick the atom a_z and for the remaining atoms, we must choose from those outside the union of $N_B(y_p)$ and $N_B(y_q)$. From $(r-(2s-1))$ elements, we can pick any $s-1$ for y. We have by assumption (M6) that

$$P(E) = \left(\frac{s}{r}\right)\frac{\binom{r-2s+1}{s-1}}{\binom{r-1}{s-1}}. \tag{10}$$

Next, we calculate

$$\begin{aligned}
\frac{\binom{r-2s+1}{s-1}}{\binom{r-1}{s-1}} &= \frac{(r-2s+1)!}{(r-3s+2)!} \cdot \frac{(r-s)!}{(r-1)!} \\
&= \frac{(r-3s+3)(r-3s+4)\dots(r-2s)(r-2s+1)}{(r-s+1)(r-s+2)\dots(r-2)(r-1)} \\
&= \left(1 - \frac{2s-2}{r-s+1}\right)\left(1 - \frac{2s-2}{r-s+2}\right)\dots\left(1 - \frac{2s-2}{r-2}\right)\left(1 - \frac{2s-2}{r-1}\right) \\
&> \left(1 - \frac{2(s-1)}{r-(s-1)}\right)^{s-1} \\
&\geq \exp\left(-\frac{2(s-1)^2}{r-(s-1)}\right).
\end{aligned}$$

The last inequality is obtained by truncating the Taylor expansion of e^{-x}.

Therefore, from Eq. (10), we can conclude that

$$P(E) \geq \frac{3s}{4r}. \tag{11}$$

For each signal y_k, where $1 \leq k \leq n$, define the random variable I_k where

$$\begin{aligned}
&I_k = 1; \quad \text{if } N_B(y_k) \cap N_B(y_p) = \{a_z\} \text{ and } N_B(y_k) \cap N_B(y_p) = \{a_z\}, \\
&I_k = 0; \quad \text{otherwise.}
\end{aligned}$$

Let $S_n = I_1 + I_2 + \cdots + I_n$ be their sum. By Eq. (11) and Lemma 1,

$$P\left(S_n < \frac{3ns}{8r}\right) \leq \exp\left(-\frac{9ns^2}{32r^2}\right).$$

If y_k belongs to the set G, it must be true that $|\langle y_k, y_p\rangle| > \tau$ and $|\langle y_k, y_p\rangle| > \tau$, for some threshold value τ. We proceed to verify that if $|N_B(y_k) \cap N_B(y_p)| = 1$, then there is a threshold value τ such that $|\langle y_k, y_p\rangle| > \tau$.

$$\begin{aligned}\left|\langle y_k, y_p\rangle\right| &= \left|\sum_i \sum_j x_i(k)x_j(p)\langle a_i, a_j\rangle\right| \\ &\geq |x_z(k)x_z(p)| \cdot \|a_z\|_2^2 - \sum_{i\neq j}\sum \left|x_i(k)x_j(p)\langle a_i, a_j\rangle\right| \\ &\geq \alpha^2 - s^2\beta^2\mu.\end{aligned}$$

Similarly, if $|N_B(y_k) \cap N_B(y_q)| = 1$, then $|\langle y_k, y_q\rangle| > \alpha^2 - s^2\beta^2\mu$.
Therefore, we can conclude that $|G| = S_n$ and the inequality (9) is valid. □

Proposition 6 rests on the assumption that we have identified a pair of signals which share exactly one atom in common. In practice, Algorithm 2 fulfills this assumption with high probability. We can now complete the proof of Theorem 1. The conclusion of Theorem 1 follows immediately from the next theorem. The two theorems are equivalent to each other.

Theorem 3 *With overwhelmingly high probability, we can nearly recover all the atoms in the dictionary. More precisely, under the general model assumptions, when there are n signals in the sample, all the atoms in the dictionary can be nearly recovered, with probability at least*

$$1 - r\exp\left(-\frac{ns^2}{2r^2}\right).$$

Proof Fix an index $u \in \{1, 2, \ldots, r\}$ and let a_u be a dictionary atom. Let Q be the set of all signals that share the atom, $Q = \{y : a_w \in N_B(y)\}$. Then, for each signal y_k, where $1 \leq k \leq n$, by assumption (M6), we have

$$P\left(y_k \in Q\right) = \frac{\binom{r-1}{s-1}}{\binom{r}{s}} = \frac{s}{r}.$$

For each $k = 1, 2, \ldots, n$, define the random variable I_k by $I_k = 1$, if $y_k \in Q$, and otherwise, $I_k = 0$. In other words, I_k is the indicator variable for whether $y_k \in Q$. By definition, $|Q| = I_1 + I_2 + \cdots + I_n$ and $E(I_j) = \frac{s}{r}$ for each j. By Lemma 1, we have

$$P\left(|Q| \leq \frac{ns}{2r}\right) \leq \exp\left(-\frac{ns^2}{2r^2}\right). \tag{12}$$

We have established that for a given atom in the dictionary, there is an overwhelmingly high probability that there is a sufficiently large cluster of signals that share that atom in common. Given a cluster of signals for that particular atom, we can recover that atom with an overwhelmingly high probability, by Propositions 5 and 6.

It remains to establish a union bound in order to prove that we can recover all the atoms in the dictionary with high probability. From Eq. (12),

$$P\left(\exists\, u \in \{1, 2, \ldots, r\} \quad \text{such that } |\{y\colon a_u \in N_B(y)\}| \leq \frac{ns}{2r}\right) \leq r \exp\left(-\frac{ns^2}{2r^2}\right).$$

which implies that

$$P\left(\forall\, u \in \{1, 2, \ldots, r\}, |\{y\colon a_u \in N_B(y)\}| \geq \frac{ns}{2r}\right) \geq 1 - r \exp\left(-\frac{ns^2}{2r^2}\right).$$

□

Finally, to see that this theorem is equivalent to Theorem 1, note that as soon as $n = O(r^2 \log(r))$, the probability becomes at least $1 - O(1/r^2)$.

7 Numerical Experiments

We conduct experiments with our data process to show that under some conditions, our modified clustering + eigenvector-based atom extraction procedure can fully recover the atoms of the original dictionary and reasonably reconstruct the original signal matrix Y without an alternating minimization step. To reconstruct the data, after the recovered dictionary D_{pred} is created by $P1$, we use the method of orthogonal matching pursuit (OMP) to form a recovered sparse matrix X_{pred}. We use OMP because it is a fast and an easily comparable baseline used widely in the literature. We perform 2 sets of experiments; when the true dictionary D is an orthogonal basis and when D is overcomplete.

7.1 Square Dictionaries

We set the following parameters of our model: $n = 2048$, $d = 256$, $s = 3$, $\beta = 10$, $\alpha = \{1, 3\}$, $\gamma = 8$, and $r = 256$. To be clear, this means that the dictionary has 256 atoms, the collection of data has 2048 signals, and each signal in $\mathbb{R}^{256}$ is 3-sparse. We remind the reader that a signal is 3-sparse means that it is a linear combination of at most 3 atoms, and setting the value of β to 10 means the largest of the three coefficients is 10. The original dictionary D for the data generating process is a Discrete Cosine Transform (DCT) matrix. We use the DCT dictionary because it is a standard choice in the literature. We choose n, d, s based on similar values used in the literature (Aharon et al. 2006), and we choose α, β, γ, and r to illustrate the reconstruction of signals. We implement each experiment in MATLAB on a computer with a Core i7-4650U processor and 8GB of RAM.

We use two metrics to judge the efficacy of our algorithm. The first is the *minimum recovery* . It is the minimum inner product that any of the recovered atoms make with a true dictionary atom, *without* taking the sign of the recovered atom into account. ν is calculated as:

$$\nu = \min_{\hat{a} \in D_{pred}} \max_{a \in D} \left| \left(||\widehat{a} - \epsilon a||_2 - 1 \right) \right|,$$

where D_{pred} is the recovered dictionary, D is the true dictionary, $|| \cdot ||_2$ is the vector L^2 norm, and $\epsilon \in \{-1, 1\}$. This metric is multiplied by 100 and reported as a percentage. Our second metric is the relative error of the reconstruction of Y, defined as

$$\text{relative reconstruction error} = \frac{||D_{pred} X_{pred} - Y||_2}{||Y||_2}.$$

where $|| \cdot ||_2$ indicates the spectral norm of a matrix (2-norm, or largest singular value), and D_{pred} and X_{pred} are, respectively, the dictionary recovered by P1, and the sparse matrix recovered by OMP against D_{pred}.

For the $\alpha = 1$ case, our correlation threshold is $\tau_1 = 2(1)(10) + (1)(1) = 21$. We run our procedure P1 to construct the dictionary, and follow it with OMP to reconstruct X. We perform this experiment five times and average the metrics below. Our algorithm scans through all possible clusters of signals and stops when it has extracted 256 atoms. It therefore has the significant benefit of determining the number of atoms in the dictionary without a priori knowledge. Each of these 256 atoms is well recovered, as $\nu = 99.91\%$. Similarly, we reconstruct the data as well, with a relative reconstruction error of 8.19%.

For the $\alpha = 3$ case, our correlation threshold is $\tau_1 = 2(3)(10) + (1)(9) = 69$. We use the same experimental setup as in the $\alpha = 1$ case. We perform this experiment five times. In all five runs, P1 stops after recovering all 256 atoms, again illustrating the automatic atom number determination that this approach enjoys. We also again recover all atoms, with $\nu = 99.31\%$. To be clear, the dictionary has 256 atoms, the collection of data has 2048 signals, and each signal in $\mathbb{R}^{256}$ is 3-sparse. In the $\alpha = 3$ case, our relative reconstruction error is 21.64%. While this is larger than the $\alpha = 1$ case, we suspected that, due to the near-perfect atom recovery rate, the error must mostly be due to the sparse coding process governed by OMP. Indeed this is the case: we found that although X_{pred} recovers almost every β-valued entry, it occasionally has flipped signs. Because in this work we mainly focus on the dictionary construction method, we do not attempt to improve this error rate; however, it is possible that it may be improved through the use of a sparse coding method more sophisticated than OMP.

7.2 Overcomplete Dictionaries

For our next set of experiments, we consider the scenario when the true dictionary D is overcomplete, i.e. when the number of columns are far greater than the number of rows.

Table 1 Parameters and minimum recovery (ν) for overcomplete dictionary recovery experiments in Sect. 7.2

d	r	β	α	s	γ	n	ν (%)
300	600	10	2	4	4	2400	98.99
300	600	10	2	5	4	2400	98.77
300	600	10	2	4	5	3000	99.06
300	600	10	2	5	5	3000	98.66
300	600	10	2	4	6	3600	99.39
300	600	10	2	5	6	3600	99.14
300	600	16	4	5	4	2400	98.14

In applications, the number of dictionary atoms usually far exceeds the dimension of each atom. We continue to assume that the dictionary has small coherence.

We start by constructing a dictionary $D \in \mathbb{R}^{300\times 600}$ by a procedure similar to the one suggested in Bandeira et al. (2017). The construction proceeds by first creating a DCT matrix of size 600 × 600, and then selecting 300 rows from it. We set $\beta = 10$, $\alpha = 2$, $s \in \{4, 5\}$, and $\gamma \in \{4, 5, 6\}$. Note that the setting of s is the tightest possible for our requirement that $r \geq 19s^3$ from the definition of ϵ_1 in Algorithm 1. We also perform one experiment with $\beta = 16$ and $\alpha = 4$. Recall that s is the sparsity parameter of our data generating process, and γ is the true cluster size parameter. For all experiments, $\epsilon_A = 0.3$ and $\mu = 0$. We calculate τ_1 according to Sect. 5. The sparse matrix X is calculated according to our data generating process, and r, the true number of atoms, is set to $600k$, so that the sparse matrix $X \in \mathbb{R}^{600\times 600k}$ and the signal matrix $Y \in \mathbb{R}^{300\times 600k}$. For this experiment, the coherence of our dictionary is orders of magnitude larger than that of a square DCT matrix, so we expect a somewhat smaller ν than in the section above. Results are reported in Table 1. We see that ν is indeed smaller, but only marginally. Almost all atoms are recovered well. The recovery rate is robust to the realistic range of γ and s.

7.3 Comparison with Existing Algorithmic Work

Our algorithm is similar in form to the initialization algorithm InitDictionaryLearn of Agarwal et al. However, our analysis under the condition of the "spiked" signal model (wherein a signal is primarily constructed from one atom, with noise from others) results in an algorithm with significantly improved ability to recover atoms. Consequently, our work is the first that we are aware of in which a dictionary initialization algorithm is capable of recovering all atoms with high probability. Existing work has applied initialization algorithms with limited success. In Agarwal et al. (2014), a comparison between the initialization algorithm and initialization followed by alternating minimization is undertaken, and the resulting recovery error incurred by initialization is 0.56 (which corresponds to $\nu = 44\%$ in our own metric). The recovery error incurred after five full rounds of alternating minimization after initialization is near zero. The authors conclude that their initialization algorithm is not sufficient to obtain an estimate of the dictionary up to reasonable accuracy. As our experimental results show, Algorithm 1 does not encounter the same challenge.

7.4 Computational Requirements

The computational complexity of Algorithm 1 is similar to that of InitDictionaryLearn. As a randomized algorithm over pairs of signals, the chief computational burden in both algorithms arises from testing the potentially large number of pairs to find "good" clusters. As such, the worst-case complexity of our algorithm is dominated by $O(n^2)$, but runs much faster in practice, as the problem is embarrassingly parallel, and our implementation takes advantage of this fact. Given the simplicity of exploiting the parallelism, we assume that Agarwal et al. take similar advantage. Although an analysis of the computational complexity of the initialization algorithm is not provided in Agarwal et al. (2014), the similar form of our algorithms implies that the dominating term is the same.

The major difference between our algorithm's computational complexity and that of Agarwal et al. lies in the work performed to solve the full dictionary learning problem. While our method is comprised of initialization followed by OMP or another sparse vector recovery procedure, Agarwal et al. require initialization followed by alternating minimization. Alternating minimization itself requires both a sparse vector recovery step and a least squares step. As the least squares step essentially relies on computing the pseudoinverse of $\mathbf{X}$, the AltMinDict algorithm of Agarwal et al. requires an additional $O(n^{2.3})$ or greater operation per iteration. Our computational savings in our dictionary learning solution are provided by our algorithm's ability to complete full atom recovery during initialization, rather than throughout both initialization and the following procedure.

8 Conclusion

Our work shows that it is indeed possible to perform dictionary learning using only a "clustering and atom extraction" initialization algorithm paired with a sparse coding algorithm, and theoretically guarantees its efficacy under our model assumptions. This allows us to bypass the requirement of running an alternating minimization procedure, and may indicate that other data generating processes enjoy this same empirical performance. We leave the construction of new data generating processes and their theoretical and empirical analysis to future research.

Acknowledgements Enrico Au-Yeung is grateful for the financial support of an Undergraduate Research Assistant Program (URAP) Grant from DePaul University. Greg Zanotti is grateful that the Undergraduate Research Assistant Program (URAP) Grant from DePaul University makes this research project possible.

References

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., Tandon, R.: Learning sparsely used overcomplete dictionaries. Proc. Mach. Learn. Res. (COLT) **35**, 123–137 (2014)

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P.: Learning sparsely used overcomplete dictionaries via alternating minimization. SIAM J. Optim. **26**(4), 2775–2799 (2016)

Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**, 4311–4322 (2006)

Arora, S., Ge, R., Moitra, A.: New algorithms for learning incoherent and overcomplete dictionaries. Proc. Mach. Learn. Res. (COLT) **35**, 779–806 (2014)

Bandeira, A.S., Mixon, D.G., Moreira, J.: A conditional construction of restricted isometries. Int. Math. Res. Not. IMRN **2**, 372–381 (2017)

Barak, B., Kelner, J.A., Steurer, D.: Dictionary learning and tensor decomposition via the sum-of-squares method. STOC (2015)

Blumensath, T., Davies, M.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–74 (2009)

Bouton, C.E., Shaikhouni, A., Annetta, N.V., Bockbrader, M.A., Friedenberg, D.A., Nielson, D.M.: e.a.: Restoring cortical control of functional movement in a human with quadriplegia. Nature **533**, 247–250 (2016)

Cai, T.T., Wang, L.: Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Trans. Inf. Theory **57**(7), 4680–4688 (2011)

Candes, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006a)

Candes, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006b)

Cohen, A., Dahmen, W., DeVore, R.: Orthogonal matching pursuit under the restricted isometry property. Constr. Approx. **45**(1), 113–127 (2017)

Daubechies, I., DeVore, R., Fornasier, M., Gunturk, S.: Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure Appl. Math. **63**(1), 1–38 (2010)

Davenport, M.A., Wakin, M.B.: Analysis of orthogonal matching pursuit using the restricted isometry property. IEEE Trans. Inf. Theory **56**(9), 4395–4401 (2010)

Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)

Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. Proc. Natl. Acad. Sci. **100**(1), 2197–2202 (2003)

Engan, K., Aase, S., Husoy, J.: Method of optimal directions for frame design. In: Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (1999)

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

Foucart, S.: Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. Approximation Theory XIII: San Antonio 2010. Springer Proceedings in Mathematics, vol. 13, pp. 65–77. Springer, New York (2012)

Gribonval, R., Schnass, K.: Dictionary identification-sparse matrix-factorization via $l1$-minimization. IEEE Trans. Inf. Theory **56**(7), 3523–3539 (2010)

Gribonval, R., Jenatton, R., Bach, F.: Sparse and spurious: dictionary learning with noise and outliers. IEEE Trans. Inf. Theory **61**(11), 6298–6319 (2015)

Hansen, F., Pedersen, G.: Jensen's operator inequality. Bull. Lond. Math. Soc. **35**(4), 553–564 (2003)

Jenatton, R., Mairal, J., Obozinski, G., Bach, F.R.: Proximal methods for sparse hierarchical dictionary learning. In: 27th International Conference on Machine Learning

Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representations for image and video restoration. Multiscale Model. Simul. **7**(1), 214–241 (2008)

Naumova, V., Schnass, K.: Dictionary learning from incomplete data for efficient image restoration. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1425–1429 (2017)

Newman, D., Shepp, L.: The double dixie cup problem. Am. Math. Mon. **67**(1), 58–61 (1960)

Schnass, K.: On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. Appl. Comput. Harmon. Anal. **37**(3), 464–491 (2014)

Spielman, D.A., Wang, H., Wright, J.: Exact recovery of sparsely-used dictionaries. Proc. Mach. Learn. Res. (COLT) **23**, 1–18 (2012)

Tosic, I., Frossard, P.: Dictionary learning. IEEE Signal Process. Mag. **28**(2), 27–38 (2011)

Tropp, J.A.: Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**(10), 2231–2242 (2004)

Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. In: Compressed Sensing, pp. 210–268. Cambridge University Press, Cambridge (2012)

Zhang, Q., Li, B.: Discriminative K-SVD for dictionary learning in face recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2691–2698 (2010)