A Fast Holistic Algorithm for Complete Dictionary Learning via ℓ^4 Norm Maximization

Yuexiang Zhai*†, Zitong Yang*, Zhenyu Liao†, John Wright‡, and Yi Ma*
*EECS, UC Berkeley †ByteDance Research Lab ‡EE, Columbia University

***Abstract*—This paper considers the problem of learning a complete (orthogonal) dictionary from sparsely generated sample signals. Unlike conventional methods that minimize ℓ^1 norm to exploit sparsity and learns the dictionary one column at a time, we propose instead to maximize ℓ^4 norm to learn the entire dictionary over the orthogonal group in a holistic fashion. We give a conceptually simple and yet effective algorithm based on matching, stretching, and projection (MSP). To justify the proposed formulation and algorithm, we study the expected behaviors of the optimization problem based on measure concentration and characterize statistically the required sample size. We also give a proof for the local convergence of the proposed MSP algorithm, as well as its superlinear (cubic) convergence rate. Experiments show that the new algorithm is significantly more efficient and effective than existing methods, including KSVD and ℓ^1-based methods. Through extensive experiments, we also show that, somewhat remarkably, maximizing ℓ^4 norm with the proposed algorithm recovers the correct dictionary under very broad conditions, well beyond current theoretical bounds.**

I. Introduction

A. Formulation and Motivations

In this work, we consider the problem of learning a complete dictionary from sparsely generated sample signals. To be more precise, an n-dimensional sample $\boldsymbol{y} \in \mathbb{R}^n$ is assumed to be a sparse superposition of columns of a complete dictionary[1] $\boldsymbol{D}_o \in \mathbb{R}^{n\times n}$: $\boldsymbol{y} = \boldsymbol{D}_o\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is a sparse (coefficient) vector. A typical statistical model for the sparse coefficient is that entries of $\boldsymbol{x}$ are iid Bernoulli Gaussian $\{x_i\} \sim_{iid} \mathrm{BG}(\theta)$[2] [2], [14], [15].

Suppose we are given a collection of sample signals $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_p] \in \mathbb{R}^{n\times p}$, each of which is generated as $\boldsymbol{y}_i = \boldsymbol{D}_o\boldsymbol{x}_i$. Write $\boldsymbol{X}_o = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p] \in \mathbb{R}^{n\times p}$. In this notation,

$$\boldsymbol{Y} = \boldsymbol{D}_o\boldsymbol{X}_o. \tag{I.1}$$

Dictionary learning is the problem of recovering both the dictionary $\boldsymbol{D}_o$ and the sparse coefficients $\boldsymbol{X}_o$, given only the samples $\boldsymbol{Y}$. Equivalently, we wish to factorize $\boldsymbol{Y}$ as $\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}$, where $\boldsymbol{D}$ is an estimate of the true dictionary $\boldsymbol{D}_o$ and $\boldsymbol{X}$ is sparse. Under our probabilistic hypotheses, the problem of learning a general complete dictionary can be reduced to that of learning an *orthogonal* dictionary[3], and so we assume without loss of generality that $\boldsymbol{D}_o$ is an orthogonal matrix: $\boldsymbol{D}_o \in \mathsf{O}(n;\mathbb{R})$.

Because $\boldsymbol{Y}$ is sparsely generated, the optimal estimate $\boldsymbol{D}_\star$ should make the associated coefficients $\boldsymbol{X}_\star$ maximally sparse. In other words, ℓ^0-norm[4] of $\boldsymbol{X}_\star$ should be as small as possible:

$$\min_{\boldsymbol{X},\boldsymbol{D}} \|\boldsymbol{X}\|_0, \quad \text{subject to} \quad \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X},\ \boldsymbol{D} \in \mathsf{O}(n;\mathbb{R}). \tag{I.2}$$

Under fairly mild conditions, globally minimizing the ℓ^0 norm recovers the true dictionary $\boldsymbol{D}_o$ [14]. But such global minimization of the ℓ^0 norm is challenging. Typically, as in the K-SVD algorithm [1], [13], one resorts to local heuristics such as orthogonal matching pursuit.[5] This approach has been widely practiced but is challenging to give guarantees. We will compare these algorithms with ours.

1) Methods based on minimizing ℓ^1 norm: Alternatively, a number of works [2], [9], [10], [14], [15] have considered the ℓ^1 norm as a convex and continuous relaxation of ℓ^0 and solved variants of the following problem instead:

$$\min_{\boldsymbol{X},\boldsymbol{D}} \|\boldsymbol{X}\|_1, \quad \text{subject to} \quad \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X},\ \boldsymbol{D} \in \mathsf{O}(n;\mathbb{R}). \tag{I.3}$$

Although ℓ^1-minimization has been widely practiced in dictionary learning, rigorous justification for its global optimality and correctness was only recently given in [15]. That work is based on the observation that, for a complete dictionary learning $\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}$, rows of $\boldsymbol{Y}$ and $\boldsymbol{X}$ span the same subspace: $\mathrm{row}(\boldsymbol{X}) = \mathrm{row}(\boldsymbol{Y})$. Hence, if $\boldsymbol{d}$ is a column of $\boldsymbol{D}$, then $\boldsymbol{d}^*\boldsymbol{Y}$ would correspond to a row[6] of $\boldsymbol{X}$, therefore highly sparse. Under certain conditions[7], one can correctly recover each of the n columns of $\boldsymbol{D}_o$ by minimizing the ℓ^1 norm of $\boldsymbol{d}^*\boldsymbol{Y}$ over a sphere:

$$\min_{\boldsymbol{d}\in\mathbb{R}^n} \|\boldsymbol{d}^*\boldsymbol{Y}\|_1, \quad \text{subject to} \quad \|\boldsymbol{d}\|_2^2 = 1. \tag{I.4}$$

Although [15] provides theoretical guarantees for the complete dictionary learning problem, it requires to solve n optimization programs of the kind (I.4) to find all n columns $\boldsymbol{d}_i$ of the desired dictionary $\boldsymbol{D}$. We will compare the latest algorithm [2] with ours.

2) Methods based on higher order norms or statistics: Our initial motivation for this work is to seek an alternative sparsity-promoting objective function that is smooth and more amenable to learning the entire dictionary in a holistic fashion over the orthogonal group $\mathsf{O}(n;\mathbb{R})$. An observation comes from the fact that over the sphere $\mathbb{S}^{n-1} \doteq \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_2 = 1\}$:

$$\max_{\boldsymbol{x}\in\mathbb{S}^{n-1}} \|\boldsymbol{x}\|_4^4 \;=\; \min_{\boldsymbol{x}\in\mathbb{S}^{n-1}} \|\boldsymbol{x}\|_0. \tag{I.5}$$

That is, global maxima of the ℓ^4 norm over the sphere are the same as minima of ℓ^0. Therefore, instead of using ℓ^1 norm in (I.4), we could promote the sparsity of $\boldsymbol{d}^*\boldsymbol{Y}$ by considering the following program:

$$\max_{\boldsymbol{d}\in\mathbb{R}^n} \|\boldsymbol{d}^*\boldsymbol{Y}\|_4^4, \quad \text{subject to} \quad \|\boldsymbol{d}\|_2^2 = 1. \tag{I.6}$$

[1] In this work, we only consider the case the dictionary is complete. See, e.g., [3], [11] for applications of complete / orthogonal dictionary learning. The more general setting in which the dictionary $\boldsymbol{D}_o$ is overcomplete is beyond the scope of this paper.

[2] I.e., each entry x_i is a product of independent Bernoulli and standard normal random variables: $x_i = \Omega_i V_i$, where $\Omega_i \sim_{iid} \mathrm{Ber}(\theta)$ and $V_i \sim_{iid} \mathcal{N}(0,1)$.

[3] As shown in [15], the general case when the dictionary $\boldsymbol{D}_o$ is not orthogonal, the problem can be converted to the orthogonal case through a preconditioning: $\boldsymbol{Y} \leftarrow \left(\frac{1}{p\theta}\boldsymbol{Y}\boldsymbol{Y}^*\right)^{-\frac{1}{2}}\boldsymbol{Y}$.

[4] The number of non-zero entries

[5] See also [12] for algorithms for learning orthogonal sparsifying transformations.

[6] In this paper, we use $\boldsymbol{d}^*$ or $\boldsymbol{D}^*$ to denote (conjugate) transpose of a vector or a matrix.

[7] Say under the Bernoulli Gaussian sparse model and with $p \geq O(n^5\log^4(n))$ samples

Unlike ℓ^0 or ℓ^1, the ℓ^4 norm is smooth so there is no reason to solve (I.6) n times separately. We can directly maximize the sum of ℓ^4 norms of all rows of $\boldsymbol{D}^*\boldsymbol{Y}$ altogether[8] while enforcing the orthogonality constraint on $\boldsymbol{D}$:

$$\max_{\boldsymbol{D}} \|\boldsymbol{D}^*\boldsymbol{Y}\|_4^4, \quad \text{subject to} \quad \boldsymbol{D} \in \mathsf{O}(n;\mathbb{R}). \tag{I.7}$$

We should note that 4th order statistical cumulant has been widely used in blind source separation or independent component analysis (ICA) since the 90's, see [7], [8] and references therein. So if $\boldsymbol{x}$ are n independent components, by finding extrema of the so-called *kurtosis*: $\text{kurt}(\boldsymbol{d}^*\boldsymbol{y}) \doteq \mathbb{E}[(\boldsymbol{d}^*\boldsymbol{y})^4] - 3\mathbb{E}[(\boldsymbol{d}^*\boldsymbol{y})^2]^2$, one can identify one independent (non-Gaussian) component x_i at a time. Algorithm wise, this is similar to using the ℓ^1 minimization (I.4) to identify one column $\boldsymbol{d}_i$ at a time for $\boldsymbol{D}$. Fast fixed-point like algorithms have been developed for this purpose [7], [8]. If $\boldsymbol{x}$ are indeed i.i.d. Bernoulli Gaussian, with $\|\boldsymbol{d}\|_2^2 = 1$, the second term in $\text{kurt}(\boldsymbol{d}^*\boldsymbol{y})$ would become a constant. The objective of ICA coincides with maximizing the sparsity-promoting ℓ^4 norm over a sphere (I.6).

The use of ℓ^4 norm can also be justified from the perspective of sum of squares (SOS). The work of [4] shows that in theory, when $\boldsymbol{x}$ is sufficiently sparse, one can utilize properties of higher order sum of squares polynomials (such as the fourth order polynomials) to correctly recover $\boldsymbol{D}$, again one column $\boldsymbol{d}_i$ at a time.

In this work, we show, however, that all the columns $\boldsymbol{d}_i$ of $\boldsymbol{D}$ should be learned together by solving the program (I.7) in a holistic fashion so that additional orthogonality constraints among all columns $\boldsymbol{d}_i^*\boldsymbol{d}_j = \delta_{ij}$ can be exploited together. As we will see, this leads to a simple algorithm which is far more efficient than existing algorithms, with working conditions well beyond those given by the theory of SOS [4] or ℓ^1 minimization [15], at least for the complete case.

Remark 1.1 (Maximizing ℓ^{2k} norm): Conceptually, to promote sparsity, one could also consider maximizing ℓ^{2k} norm[9] of $\boldsymbol{D}^*\boldsymbol{Y}$ for any $k \geq 2$. Most analysis and results given in this paper would extend to these general cases. Nevertheless, as we will discuss and see, the case $2k = 4$ strikes a good balance between sample size and convergence rate.

B. When is the Problem Well Posed?

Notice that in the dictionary learning problem (I.1), there are some intrinsic ambiguities regarding recovering $\boldsymbol{D}_o$ in a holistic fashion: given any signed permutation matrix $\boldsymbol{P} \in \text{SP}(n)$,[10] we have:

$$\boldsymbol{Y} = \boldsymbol{D}_o\boldsymbol{X}_o = \boldsymbol{D}_o\boldsymbol{P}\boldsymbol{P}^*\boldsymbol{X}_o,$$

where $\boldsymbol{P}^*\boldsymbol{X}_o$ is equally sparse as $\boldsymbol{X}_o$. So we can only expect to best recover the correct dictionary (and sparse coefficient) *up to an arbitrary signed permutation.* Due to this ambiguity, we claim the ground truth dictionary $\boldsymbol{D}_o$ is successfully recovered, if any *signed permuted* version $\boldsymbol{D}_o\boldsymbol{P}$ is found.

C. Main Results and Contributions

1) Statistical justification: Suppose that our signal matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ is randomly generated from (I.1), we claim that the expected behaviors of solving the following ℓ^4 norm maximization problem:

$$\max_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \tag{I.8}$$

are largely characterized by the following (deterministic) program:

$$\max_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \tag{I.9}$$

whose global optima are $\boldsymbol{D}_o^*$ up to arbitrary signed permutations (that is, $\boldsymbol{A}_\star$ shall satisfy $\boldsymbol{A}_\star\boldsymbol{D}_o \in \text{SP}(n)$). We provide some simple statistical conditions and justifications why this is the case.

2) A holistic fast optimization algorithm: Unlike almost all previous algorithms that find the dictionary one column $\boldsymbol{d}_i$ at a time, we introduce a novel *matching, stretching, and projection* (MSP) algorithm that solves the programs (I.8) and (I.9) directly for the entire $\boldsymbol{D} \in \mathsf{O}(n;\mathbb{R})$. The algorithm exploits the statistics of ℓ^4 and global geometry of $\mathsf{O}(n;\mathbb{R})$ to achieve a cubic convergence rate. Extensive experiments show that the algorithm is far more efficient than existing heuristic or (Riemannian) gradient or subgradient based algorithms. With this efficient algorithm, we characterize the range of success for the program (I.8), which goes well beyond any existing theoretical guarantees [4], [15] for the complete dictionary case.

D. Notations

We use a bold uppercase and lowercase letters to denote matrices and vectors respectively: $\boldsymbol{X} \in \mathbb{R}^{n\times p}, \boldsymbol{x} \in \mathbb{R}^n$. We reserve small letter for scalar: $x \in \mathbb{R}$. We use $\|\boldsymbol{X}\|_4$ to denote the element-wise ℓ^4 norm of matrix $\boldsymbol{X}$. We use $\boldsymbol{D}_o$ to denote the ground truth dictionary, and $\boldsymbol{A}$ is an estimate for $\boldsymbol{D}_o^*$ from solving (I.8). We use $\circ$ to denote the Hadamard product: $\forall \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\{\boldsymbol{A}\circ\boldsymbol{B}\}_{i,j} = a_{i,j}b_{i,j}$, and $\{\boldsymbol{A}^{\circ r}\}_{i,j} = a_{i,j}^r$ is the element-wise r^{th} power of $\boldsymbol{A}$.

Given an input data matrix $\boldsymbol{Y}$ randomly generated from (I.1), for any orthogonal matrix $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$, we define $\hat{f} : \mathsf{O}(n;\mathbb{R}) \times \mathbb{R}^{n\times p} \mapsto \mathbb{R}$ as the 4th power of ℓ^4 norm of $\boldsymbol{A}\boldsymbol{Y}$:

$$\hat{f}(\boldsymbol{A},\boldsymbol{Y}) \doteq \|\boldsymbol{A}\boldsymbol{Y}\|_4^4. \tag{I.10}$$

We define $f : \mathsf{O}(n;\mathbb{R}) \mapsto \mathbb{R}$ as the expectation of $\hat{f}$ over $\boldsymbol{X}_o$:

$$f(\boldsymbol{A}) \doteq \mathbb{E}_{\boldsymbol{X}_o}[\hat{f}(\boldsymbol{A},\boldsymbol{Y})] = \mathbb{E}_{\boldsymbol{X}_o}\big[\,\|\boldsymbol{A}\boldsymbol{Y}\|_4^4\,\big]. \tag{I.11}$$

For any orthogonal matrix $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$, we define $g : \mathsf{O}(n;\mathbb{R}) \mapsto \mathbb{R}$ as 4^{th} power of its ℓ^4 norm: $g(\boldsymbol{A}) \doteq \|\boldsymbol{A}\|_4^4$.

E. Organization of the Paper

Rest of the paper is organized as follows. In Section II, we characterize the global maximizers of (I.8) statistically via measure concentration. In Section III-B, we describe the proposed MSP algorithm, and in Section III-C, we characterize fixed points of the algorithm and show its local convergence rate. *Due to space limit, we leave all proofs to the full version of the paper.* Finally, in Section IV, we conduct extensive experiments to show effectiveness and efficiency of our method, by comparing with the state of the art.

II. Statistical Justification

In this section, we provide some basic statistical justification for why we would expect the program

$$\max \hat{f}(\boldsymbol{A},\boldsymbol{Y}) = \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}) \tag{II.1}$$

to recover the ground truth dictionary $\boldsymbol{D}_o$.

- Firstly, we will show that statistically the (random) function $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ concentrates on its expectation $f(\boldsymbol{A})$ as p is polynomial in n (Lemma 2.1).
- Secondly, the expectation $f(\boldsymbol{A})$ is a linear function of $g(\boldsymbol{A}\boldsymbol{D}_o) = \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4$, and as result they have the same global maxima (Lemma 2.2).
- Finally, we show that all global maxima of $g(\boldsymbol{A}\boldsymbol{D}_o)$ are signed permutations of $\boldsymbol{D}_o$ (Lemma 2.3).

[8] meaning the sum of 4^{th} powers of all entries of a matrix: $\forall \boldsymbol{A} \in \mathbb{R}^{n\times m}$ $\|\boldsymbol{A}\|_4^4 = \sum_{i,j} a_{i,j}^4$.

[9] The "ℓ^{2k} norm" of a matrix is the sum of $2k^{\text{th}}$ power of all of its entries: $\forall \boldsymbol{A} \in \mathbb{R}^{n\times m}$ $\|\boldsymbol{A}\|_{2k}^{2k} = \sum_{i,j} a_{i,j}^{2k}$.

[10] $\text{SP}(n)$ here denotes the group of signed permutation matrices, more specifically, orthogonal matrices only contain $0, \pm 1$.

Lemma 2.1 (Concentration of $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$): $\forall \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \forall \varepsilon > 0$, $\frac{1}{np}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ has the following concentration bound to its expectation $\frac{1}{np}f(\boldsymbol{A})$:

$$\mathbb{P}\left(\left|\frac{\hat{f}(\boldsymbol{A},\boldsymbol{Y})}{np} - \frac{f(\boldsymbol{A})}{np}\right| > \varepsilon\right) \le O\Big(\frac{n^4\theta^4}{p\varepsilon^2}\Big). \tag{II.2}$$

The proof is based on Chebyshev inequality and we leave the details in the full version of the paper.

This indicates that the (random) function $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ behaves like its expectation $f(\boldsymbol{A})$ as the sample size p increases. For this approximation to be good with high probability, the number of samples p only need to be polynomial in n and $1/\epsilon$, or more precisely $p > O(n^4/\epsilon^2)$.

Due to concentration, to large extent, the properties of maximizing $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ can be studied through examining how the deterministic function $f(\boldsymbol{A})$ can be optimized:

$$\max_{\boldsymbol{A}} f(\boldsymbol{A}) = \mathbb{E}_{\boldsymbol{X}_o}\big[\|\boldsymbol{A}\boldsymbol{Y}\|_4^4\big], \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}). \tag{II.3}$$

Lemma 2.2 (Properties of $f(\boldsymbol{A})$): $\forall \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$ and $\forall \theta \in (0,1)$, $f(\boldsymbol{A})$ has the following properties:

- $\frac{1}{3p\theta}f(\boldsymbol{A}) = (1-\theta)g(\boldsymbol{A}\boldsymbol{D}_o) + \theta n$.
- $\frac{1}{3p\theta}f(\boldsymbol{A}) \le n$, with equality holds if and only if $\boldsymbol{A}\boldsymbol{D}_o \in \mathrm{SP}(n)$.

This lemma shows that $f(\boldsymbol{A})$ and $g(\boldsymbol{A}\boldsymbol{D}_o)$ are linearly related hence their global maxima are the same on $\mathsf{O}(n;\mathbb{R})$:

$$\boldsymbol{A} = \operatorname*{arg\,max}_{\boldsymbol{A}\in\mathsf{O}(n,\mathbb{R})} f(\boldsymbol{A}) \text{ if and only if } \boldsymbol{A} = \operatorname*{arg\,max}_{\boldsymbol{A}\in\mathsf{O}(n,\mathbb{R})} g(\boldsymbol{A}\boldsymbol{D}_o).$$

Thus, maximizing $f(\boldsymbol{A})$ is equivalent to the following optimization problem:

$$\max_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o) = \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}), \tag{II.4}$$

Moreover, the extrema of $g(\cdot)$ on $\mathsf{O}(n;\mathbb{R})$ are well structured, the following lemma makes this precise.

Lemma 2.3 (Extrema of ℓ^4 Norm on the Orthogonal Group): For any $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$, $g(\boldsymbol{A}) = \|\boldsymbol{A}\|_4^4 \in [1,n]$, $g(\boldsymbol{A})$ reaches maximal value n if and only if $\boldsymbol{A} \in \mathrm{SP}(n)$ and $g(\boldsymbol{A})$ reaches minimum if and only if $\boldsymbol{A}$ is a Hadamard matrix[11].

This lemma implies that if $\boldsymbol{A}_\star$ is a maximum of $g(\boldsymbol{A}\boldsymbol{D}_o)$, i.e. $\|\boldsymbol{A}_\star\boldsymbol{D}_o\|_4^4 = n$, then it differs from $\boldsymbol{D}_o$ by a signed permutation.

The following lemma shows that when the ℓ^4 norm of an orthogonal matrix $\boldsymbol{A}$ is close to the maximum value n, it is also close to a signed permutation matrix in Frobenius norm.

Lemma 2.4 (Extrema of ℓ^4 norm over the orthogonal group): Suppose $\boldsymbol{A}$ is an orthogonal matrix: $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$. For arbitrarily small ε, if $\frac{1}{n}\|\boldsymbol{A}\|_4^4 > 1-\varepsilon$, then $\exists \boldsymbol{P} \in \mathrm{SP}(n)$, such that

$$\frac{1}{n}\|\boldsymbol{A}-\boldsymbol{P}\|_F^2 < C_1 n\varepsilon. \tag{II.5}$$

This result is useful whenever we try to evaluate how close a solution from an algorithm is to the optimal one.

Remark 2.5 (Maximizing ℓ^{2k} norm): If one were to choose maximizing ℓ^{2k} norm to promoting sparsity, similar analysis of concentration bounds would reveal that for the same error bound, it requires much larger number p of samples for the (random) objective function $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ to concentrate on its (deterministic) expectation $f(\boldsymbol{A})$. Experiments in Section IV-E corroborate with the findings.

[11]Note that there is no guarantee that Hadamard matrix exists $\forall n \in \mathbb{N}^+$, [16] shows that Hadamard matrix exists for infinite many n.

III. Matching, Stretching, and Projection Algorithm

In this section, we introduce an algorithm, based on a simple iterative *matching, stretching, and projection* (MSP) process, which efficiently solves the two related programs (I.8) and (II.4).

A. Related Optimization Methods

Although (I.8) is everywhere smooth, the associated optimization is non-trivial in several ways. First, one needs to deal with the signed permutation ambiguity. The problem has combinatorially many global maximizers. Furthermore, we are maximizing a convex function (or minimizing a concave function) over a constraint set. So conventional methods such as augmented Lagrangian barely works. This is because the Lagrangian [6]: $\mathcal{L}(\boldsymbol{A},\boldsymbol{\Lambda}) \doteq -\|\boldsymbol{A}\boldsymbol{Y}\|_4^4 + \langle \boldsymbol{A}\boldsymbol{A}-\boldsymbol{I},\boldsymbol{\Lambda}\rangle$ will go to negative infinity due to the concavity of the objective function $-\|\boldsymbol{A}\boldsymbol{Y}\|_4^4$. Notice that all of its global maximizers are on the constraint set, an ideal iterative algorithm should converge to a solution that *exactly lies on constraint set* $\mathsf{O}(n;\mathbb{R})$.

Another natural way to optimize (I.8) is to apply Riemannian gradient (or projected gradient) type methods [5] on the group $\mathsf{O}(n;\mathbb{R})$. One can take small gradient steps to ensure convergence. Such methods converge at best with a linear rate (if the objective function is strongly convex). Nevertheless, as we will see, due to special global geometry of the problem, we can choose a very large (even infinite!) step size and the process converges much more rapidly.

B. ℓ^4 Maximization over $\mathsf{O}(n;\mathbb{R})$ via an MSP Algorithm

We now introduce the matching, stretching and projection (MSP) algorithm to solve problems (I.8) and (II.4). Meanwhile, we also provide analysis and justification why the proposed algorithm is expected to work well.

a) The Deterministic Case: Since the dictionary learning optimization problem (I.8) concentrates on the ℓ^4 norm maximization problem (II.4) w.h.p., we first introduce our MSP algorithm for solving (II.4):

$$\max_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o) = \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}).$$

Algorithm 1 MSP for ℓ^4 Maximization over Orthogonal Group

1: **Given any** $\boldsymbol{D}_o \in \mathsf{O}(n,\mathbb{R})$. ▷ Ground truth $\boldsymbol{D}_o$
2: **Initialize**: $\boldsymbol{A}_0 \in \mathsf{O}(n,\mathbb{R})$. ▷ Initialize $\boldsymbol{A}_0$ for iteration
3: **for** $t = 0, 1, \ldots$ **do**
4: $\quad \partial\boldsymbol{A}_t = 4(\boldsymbol{A}_t\boldsymbol{D}_o)^{\circ 3}\boldsymbol{D}_o^*$; ▷ $\nabla_{\boldsymbol{A}}\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 = 4(\boldsymbol{A}\boldsymbol{D}_o)^{\circ 3}\boldsymbol{D}_o^*$
5: $\quad \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \mathrm{svd}(\partial\boldsymbol{A}_t)$;
6: $\quad \boldsymbol{A}_{t+1} = \boldsymbol{U}\boldsymbol{V}^*$; ▷ Project $\boldsymbol{A}$ onto orthogonal group
7: **end for**
8: **Output**: $\boldsymbol{A}_{t+1}, \|\boldsymbol{A}_{t+1}\boldsymbol{D}_o\|_4^4/n$.

Note that in the output we normalize $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4$ by dividing n, because the global maximum of $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4$ is n and the output is therefore normalized to 1. In Step 4 of the MSP algorithm, the calculation of $\partial\boldsymbol{A}_t = 4(\boldsymbol{A}_t\boldsymbol{D}_o)^{\circ 3}\boldsymbol{D}_o^*$ does not require knowledge of $\boldsymbol{D}_o$. It is merely the gradient of the objective function

$$\nabla_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o) = \nabla_{\boldsymbol{A}}\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4 = 4(\boldsymbol{A}\boldsymbol{D}_o)^{\circ 3}\boldsymbol{D}_o^*.$$

However, one shall not mistaken the MSP algorithm as a gradient descent type algorithm. In fact, in Step 4, the scale of $\partial\boldsymbol{A}_t$ is very large and the iterates are *not* incremental local updates. Due to the

scale invariant of SVD[12], one can even scale $\partial \boldsymbol{A}_t$ arbitrarily large and the algorithm still converges!

As the name of the algorithm suggests, each iteration actually performs a "matching, stretching, and projection" operation. It first matches the current estimate $\boldsymbol{A}_t$ with the true $\boldsymbol{D}_o$. Then the element-wise cubic function $(\cdot)^{\circ 3}$ stretches all entries of $\boldsymbol{A}_t\boldsymbol{D}_o$ by promoting the large ones and suppressing the small ones. $\partial \boldsymbol{A}_t$ is the correlation between so "sparsified" pattern and the original basis $\boldsymbol{D}_o^*$, which is then projected back onto the closest orthogonal matrix $\boldsymbol{A}_{t+1}$ in Frobenius norm.

Repeating this "matching, stretching, and projection" process, $\boldsymbol{A}_t\boldsymbol{D}_o$ is increasingly sparsified while ensuring the orthogonality of $\boldsymbol{A}_t$. Ideally the process will stop when $\boldsymbol{A}_t\boldsymbol{D}_o$ becomes the sparsest, that is, a signed permutation matrix. Since the iterative MSP algorithm utilizes the global geometry of the orthogonal group and acts more like the *power iteration* method or the fixed point algorithm [8], our analysis will show that it achieves *super-linear* convergence.

b) The Random Case: For the original dictionary learning problem (I.8):

$$\max_{\boldsymbol{A}} \hat{f}(\boldsymbol{A},\boldsymbol{Y}) = \|\boldsymbol{A}\boldsymbol{Y}\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}),$$

we could propose a similar "matching, stretching, and projection" (MSP) algorithm:

Algorithm 2 MSP for ℓ^4 Maximization Based Dictionary Learning

1: **Given**: $\boldsymbol{Y} = \boldsymbol{D}_o\boldsymbol{X}_o \in \mathbb{R}^{n\times p}$. ▷ $\boldsymbol{D}_o \in \mathsf{O}(n,\mathbb{R}), \boldsymbol{X}_o \sim_{iid} \mathrm{BG}(\theta)$
2: **Initialize**: $\boldsymbol{A}_0 \in \mathsf{O}(n,\mathbb{R})$. ▷ Initialize $\boldsymbol{A}_0$ for iteration
3: **for** $t = 0, 1, \ldots$ **do**
4: $\quad \partial \boldsymbol{A}_t = 4(\boldsymbol{A}_t\boldsymbol{Y})^{\circ 3}\boldsymbol{Y}^*$; ▷ $\nabla_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{Y}\|_4^4 = 4(\boldsymbol{A}\boldsymbol{Y})^{\circ 3}\boldsymbol{Y}^*$
5: $\quad \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \mathrm{svd}(\partial \boldsymbol{A}_t)$;
6: $\quad \boldsymbol{A}_{t+1} = \boldsymbol{U}\boldsymbol{V}^*$; ▷ Project $\boldsymbol{A}$ onto orthogonal group
7: **end for**
8: **Output**: $\boldsymbol{A}_{t+1}, \|\boldsymbol{A}_{t+1}\boldsymbol{Y}\|_4^4/3np\theta, \|\boldsymbol{A}_{t+1}\boldsymbol{D}_o\|_4^4/n$.

Note that in the output we also normalize $\|\boldsymbol{A}\boldsymbol{Y}\|_4^4$ by dividing the maximum of its expectation: $3np\theta$ so that the optimal output value would be around 1.

The same intuition of "matching, stretching, and projection" for the deterministic case naturally carries over here. Here in Step 4, the estimate $\boldsymbol{A}_t$ is first matched with the observation $\boldsymbol{Y}$. The cubic function $(\cdot)^{\circ 3}$ re-scales the results and promotes entry-wise sparsity of $\boldsymbol{X}_t = \boldsymbol{A}_t\boldsymbol{Y}$ accordingly. Again, here $\partial \boldsymbol{A}_t$ is the gradient $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ of the objective function, but because of its large scale, the algorithm is not performing gradient descent.

Although the data and the objective function are random here, Lemma 3.1 below shows that $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ concentrates on its expectation when p increases. Theorem 3.2 further shows its linear relationship with $\nabla_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o)$.

Lemma 3.1 (Concentration Bound of $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$): Under the same assumption of $\boldsymbol{A},\boldsymbol{Y}$ as (I.1), $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ concentrates to its expectation with the following bound

$$\mathbb{P}\left(\left\| \frac{1}{p}\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y}) - \frac{1}{p}\mathbb{E}_{\boldsymbol{X}_o}\left[\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})\right] \right\|_F^2 > \varepsilon \right) \le O\Big(\frac{n^7\theta^4}{p\varepsilon}\Big).$$

Theorem 3.2 (Expectation of $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$): With $\boldsymbol{Y}$ defined as (I.1), the expectation of $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ satisfies:

$$\mathbb{E}_{\boldsymbol{X}_o}\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y}) = 3p\theta(1-\theta)\nabla_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o) + 12p\theta^2\boldsymbol{A}. \quad \text{(III.1)}$$

[12] $\forall \boldsymbol{A} \in \mathbb{R}^{n\times n}, \alpha > 0$, the rotation matrices $\boldsymbol{U},\boldsymbol{V}$ from its SVD is the same as the $\boldsymbol{U},\boldsymbol{V}$ of $\alpha\boldsymbol{A}$.

Notice that the second term of (III.1) can be viewed as an offset between the expected gradient and the gradient of $g(\boldsymbol{A}\boldsymbol{D}_o)$ at $\boldsymbol{A}$. When θ is small (i.e. $\boldsymbol{X}_o$ sufficiently sparse), the expected gradient of $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ aligns well with that of $g(\boldsymbol{A}\boldsymbol{D}_o)$.

With these results, the direction of $\mathbb{E}_{\boldsymbol{X}_o}\left[\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})\right]$ is a linear combination of $\nabla_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o)$ and $\boldsymbol{A}$. So we expect the stretching $\partial \boldsymbol{A}_t = 4(\boldsymbol{A}_t\boldsymbol{Y})^{\circ 3}\boldsymbol{Y}^*$ in Step 4 of Algorithm 2 also promotes the sparsity of $\boldsymbol{A}_t\boldsymbol{D}_o$ w.h.p., as long as $\theta \in (0,1)$. Moreover, the stretching direction of $\nabla_{\boldsymbol{A}}\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ approximates $\nabla_{\boldsymbol{A}} g(\boldsymbol{A}\boldsymbol{D}_o)$ better with smaller θ (sparser $\boldsymbol{X}_o$), which suggests that the learning algorithm is more likely to succeed with sparser $\boldsymbol{X}_o$, as will be verified by the experiments.

C. Convergence Analysis of the MSP Algorithm

In this section, we provide convergence analysis of the proposed MSP Algorithm 1 over the orthogonal group. Notice that the objective function $g(\cdot)$ is invariant over $\mathsf{O}(n;\mathbb{R})$. So without loss of generality, we only need to provide convergence analysis for the case $\boldsymbol{D}_o = \boldsymbol{I}$.

When $\boldsymbol{D}_o = \boldsymbol{I}$, we want to show the MSP algorithm converges to a signed permutation matrix for the optimization problem:

$$\max_{\boldsymbol{A}} g(\boldsymbol{A}) = \|\boldsymbol{A}\|_4^4, \quad \text{subject to} \quad \boldsymbol{A} \in \mathsf{O}(n;\mathbb{R}),$$

starting from any initial $\boldsymbol{A}_0$ on $\mathsf{O}(n;\mathbb{R})$. For this purpose, we first introduce some basic properties of our objective function $g(\cdot)$. It is easy to show that all critical points $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ of $\|\boldsymbol{W}\|_4^4$ on the manifold $\mathsf{O}(n;\mathbb{R})$ satisfy the following condition:

$$(\boldsymbol{W}^{\circ 3})^*\boldsymbol{W} = \boldsymbol{W}^*\boldsymbol{W}^{\circ 3}. \quad \text{(III.2)}$$

The following lemma shows that all real solutions of the algebraic equations (III.2) are discrete.

Lemma 3.3 (Discreteness of Critical Points): The set of critical points of ℓ^4 norm over $\mathsf{O}(n;\mathbb{R})$ is discrete. That is, solutions to the algebraic equations:

$$(\boldsymbol{W}^{\circ 3})^*\boldsymbol{W} = \boldsymbol{W}^*\boldsymbol{W}^{\circ 3}, \quad \boldsymbol{W}^*\boldsymbol{W} = \boldsymbol{I}, \quad \text{(III.3)}$$

are discrete.

One can also give good bounds on the number of critical points using tools from algebraic geometry such as Bézout's theorem [17]. For now we have the following relation between fixed points of the MSP Algorithm 1 (when $\boldsymbol{D}_o = \boldsymbol{I}$) and the critical points of $g(\cdot)$:

Lemma 3.4 (Fixed Points of MSP): $\forall \boldsymbol{W} \in \mathsf{O}(n;\mathbb{R})$, $\boldsymbol{W}$ is a fix point of the MSP algorithm if only if $\boldsymbol{W}$ is a critical point of the ℓ^4 norm over $\mathsf{O}(n;\mathbb{R})$.

Although the function $g(\boldsymbol{W}) = \|\boldsymbol{W}\|_4^4$ may have many critical points, the signed permutation group $\mathrm{SP}(n)$ are the only global maximizers. As recent work has shown [15], such discrete symmetry helps regulate the global landscape of the objective function and makes it amenable to global optimization. Indeed, we have observed through our extensive experiments that, under broad conditions, the proposed MSP algorithm always converges to the globally optimal solution (set), at a super-linear convergence rate.

In this paper, we give a local result on the convergence of the MSP algorithm.[13] That is, when the initial orthogonal matrix $\boldsymbol{A}$ is "close" enough to a signed permutation matrix, the MSP algorithm converges to that signed permutation at a very fast rate. It is easy to verify the algorithm is permutation invariant. Hence w.l.o.g., we may assume the target signed permutation is the identity $\boldsymbol{I}$.

[13] We leave the study of ensuring global optimality and convergence to future work.

Theorem 3.5 (Local Convergence Rate of the MSP Algorithm): Given $\boldsymbol{A} \in \mathsf{O}(n;\mathbb{R})$, if $\|\boldsymbol{A}-\boldsymbol{I}\|_F^2 = \varepsilon$, and let $\boldsymbol{A}'$ denote the output of the MSP Algorithm 1 after one iteration: $\boldsymbol{A}' = \boldsymbol{U}\boldsymbol{V}^*$, where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \mathrm{svd}(\boldsymbol{A}^{\circ 3})$, then $\|\boldsymbol{A}'-\boldsymbol{I}\|_F^2 \le O(\varepsilon^3)$.

Theorem 3.5 shows that the MSP Algorithm 1 achieves cubic convergence rate locally, which is much faster than any gradient descent methods. Our experiments in Section IV confirm this super-linear convergence rate for the MSP algorithms.

Remark 3.6 (Maximizing ℓ^{2k} norm): One can easily extend the MSP algorithm to maximize ℓ^{2k} norm over the orthogonal group. In fact, the resulting algorithm would have a higher rate of convergence for the deterministic case, as the stretching with the power $(\cdot)^{\circ 2k-1}$ sparsifies the matrix more significantly with a larger k.[14] See Section IV-E for experimental verification. However, as we discussed in the previous section, for the random case, the number of samples required would increase drastically. Also see Section IV-E for experiments. The choice of $2k = 4$ seems to be the best in terms of balancing these two contending factors.

IV. Experiments

In this section, we first compare our algorithm with the classic heuristic KSVD algorithm [1] and the latest provably correct dictionary learning method [2] based on minimizing ℓ^1 norm via subgradient. We conduct additional experiments to reveal surprising performance and working ranges of the MSP algorithm, well beyond our current analysis.

A. Comparison with Prior Work

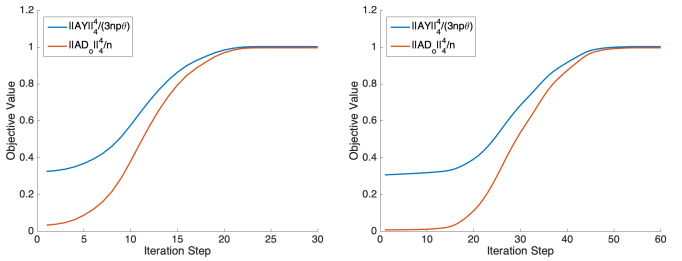
Table I below compares the MSP method with the KSVD [1] and the latest subgradient method [2] for different choices of n, p under the same sparsity level $\theta = 0.3$. As one may see, our algorithm is significantly faster than both algorithms in all trails. Further more, our algorithm has the potential for large scale experiments: it only takes 374.2 seconds to learn a 400×400 dictionaries from $160,000$ samples. While the previous algorithms either fail to find the correct dictionary or barely applicable. Within statistical errors, our algorithm gives slightly smaller values for $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n$ in some trails. But the subgradient method [2] uses information of the ground truth dictionary $\boldsymbol{D}_o$ in their stopping criteria. Our MSP algorithm removes this dependency with only mild loss in accuracy.

	KSVD [1]		Subgradient [2]		MSP (Ours)	
Trails	Error	Time	Error	Time	Error	Time
(a)	12.35%	51.2s	0.27%	35.6s	**0.34%**	**0.4s**
(b)	8.63%	244.4s	0.28%	354.9s	**0.34%**	**1.5s**
(c)	6.15%	684.9s	1.28%	6924.6s	**0.35%**	**7.6s**
(d)	8.61%	1042.3s	N/A	$>$ 12h	**0.35%**	**48.0s**
(e)	13.07%	5401.9s	N/A	$>$ 12h	**0.35%**	**374.2s**

Table I: Comparison experiments with [1], [2] in different trails of dictionary learning: (a) $n = 25, p = 1\times 10^4, \theta = 0.3$; (b) $n = 50, p = 2\times 10^4, \theta = 0.3$; (c) $n = 100, p = 4\times 10^4, \theta = 0.3$; (d) $n = 200, p = 4\times 10^4, \theta = 0.3$; (e) $n = 400, p = 16\times 10^4, \theta = 0.3$. $\boldsymbol{Y}$ is generated from (I.1). Recovery error is measured as $\left|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n\right|$, since Lemma 2.3 shows that a perfect recovery gives $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n = 1$. All experiments are conducted on a 2.7 GHz Intel Core i5 processor (CPU of a 13-inch Mac Pro 2015).

B. Dictionary Learning Convergence Rate Plot

Figure 1(a) presents one trial of the proposed MSP Algorithm 2 for dictionary learning with $\theta = 0.3$, $n = 100$, and $p = 40,000$. The result corroborates with statements in Lemma 2.1 and Lemma 2.2: maximizing $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ is largely equivalent to optimizing $g(\boldsymbol{A}\boldsymbol{D}_o)$, and both values reach global maximum at the same time. Meanwhile, this result also shows our MSP algorithm is able to find the global maximum at ease, since $g(\boldsymbol{A}\boldsymbol{D}_o)$ reaches its maximal value 1 (with minor errors) by maximizing $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$. In Figure 1(b), we test the MSP Algorithm 2 in higher dimension $n = 400, p = 1.6\times 10^5, \theta = 0.3$. In both cases, our algorithm is surprisingly efficient: it only takes around 20 iterations to recover a 100-dimensional dictionary and 50 iterations for a 400-dimensional dictionary.

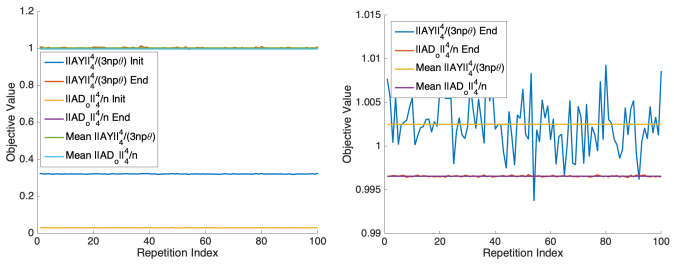


(a) $n = 100, p = 40,000, \theta = 0.3$ (b) $n = 400, p = 1.6\times 10^5, \theta = 0.3$

Fig. 1: Normalized objective value $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/3np\theta$ and $\|\boldsymbol{A}\boldsymbol{Y}\|_4^4/n$ for individual trails of the MSP Algorithm 2, with different parameters n, p, θ. According to Lemma 2.3, $g(\boldsymbol{A}\boldsymbol{D}_o)/n$ reaches 1 indicates successful recovery for $\boldsymbol{D}_o$. This experiment shows the MSP algorithm finds global maxima of $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ thus recovers the correct dictionary $\boldsymbol{D}_o$.

C. Multiple Trials of the MSP Algorithm for Dictionary Learning

In Figure 2, we run the MSP Algorithm 2 with $n = 100, p = 40,000, \theta = 0.3$ for 100 trails. Among all 100 trails, $g(\boldsymbol{A}\boldsymbol{D}_o)$ achieve the global maximal value (within statistical errors) via optimizing $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ in less than 30 iterations. This experiment seems to support a conjecture: within conditions of this experiment, the MSP algorithm recovers the globally optimal dictionary.



(a) Plot with Initial Values (b) Plot without Initial Values

Fig. 2: Normalized initial and final objective values of $\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/3np\theta$ and $\|\boldsymbol{A}\boldsymbol{Y}\|_4^4/n$ for 100 trails of the MSP Algorithm 2, with $n = 100, p = 40,000, \theta = 0.3$. Both $\hat{f}(\boldsymbol{A},\boldsymbol{Y})$ and $g(\boldsymbol{A}\boldsymbol{D}_o)$ converge to 1 (with minor errors) for all 100 trails.

D. Working Ranges of the MSP Algorithm

Encouraged by previous experiment, we conduct more extensive experiments of the MSP Algorithm 2 in broader settings to find its working range: 1) Figure 3 shows the result of varying the sparsity level θ and sample size p with a fixed dimension n and 2) Figure 4 shows results of changing dimension n and sample size p at a fixed sparsity level $\theta = 0.5$. Notice that both figures demonstrate a clear phase transition for the working range. It is somewhat surprising to see in Figure 3 that the MSP algorithm is able to recover the dictionary correctly up to the sparsity level of $\theta \approx 0.6$ if p is large

[14] In fact, one can show that if $2k \to \infty$, the corresponding MSP algorithm converges with *only one* iteration for the deterministic case!

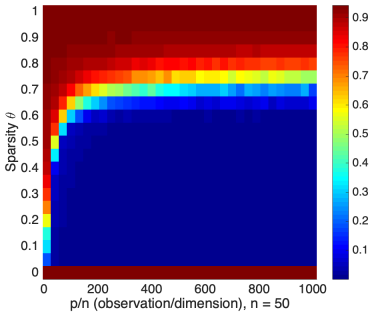
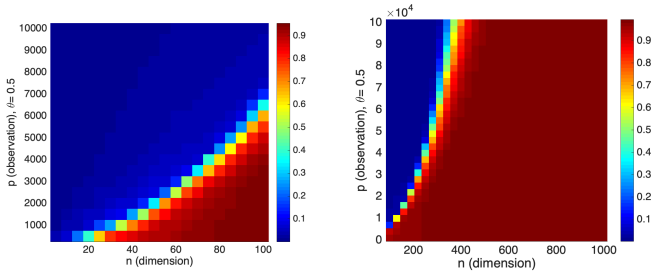


Fig. 3: Phase transition plot of average normalized error $|1-\|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n|$ among 10 trails of the MSP Algorithm 2 with $n = 50$, varying θ from 0 to 1, and p from 0 to 10,000. Red area indicates large error and blue area small error.

enough, which almost doubles the best existing theoretical guarantee given in [2], [15].

Figure 4(a) and (b) show the working range for varying n, p with a fixed $\theta = 0.5$. Figure 4(a) is for a smaller range of n (from 10 to 100) and Figure 4(b) for a larger range of n (from 100 to 1,000). It can be seen from these figures that the required sample size p for the algorithm to succeed seems to be quadratic in the dimension n: $p = O(n^2)$. This empirical bound is significantly better than the best theoretical bounds given in [2], [15] and our analysis, where at least $p = \Omega(n^4)$ samples are required to ensure success. Similar empirical observations have been reported in [2], which together suggest better analysis might be necessary to tighten the bounds.



(a) Changing n from 10 to 100 and p from 1,000 to 10,000, $\theta = 0.5$.

(b) Changing n from 100 to 1,000 and p from 10,000 to 100,000, $\theta = 0.5$.

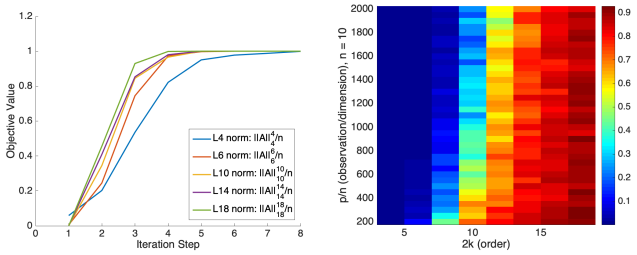
Fig. 4: Phase transition of average normalized error $|1 - \|\boldsymbol{A}\boldsymbol{D}_o\|_4^4/n|$ of 10 trails for the MSP Algorithm 2 at $\theta = 0.5$, varying n and p. Red area indicates large error and blue area small error.

E. Generalization to ℓ^{2k} Norm

In Figure 5, we conduct experiments to support the choice of ℓ^4 norm. Figure 5(a) shows that for the deterministic case, the MSP Algorithm 1 finds signed permutation matrices faster with higher order ℓ^{2k} norm. But Figure 5(b) indicates that as the order $2k$ increases, much more samples are needed by Algorithm 2 to achieve the same estimation error: p grows drastically as k increases. Hence, among all these sparsity-promoting norms (ℓ^{2k}), the ℓ^4 norm strikes a good balance between sample size and convergence rate.

V. Summary and Future Work

Dictionary learning has become an increasingly important and powerful tool in unsupervised learning for data analysis. In this paper, we see that a complete sparsifying dictionary can be learned very effectively in a holistic fashion by the simple MSP algorithm with superlinear convergence rate. The new algorithm exploits higher (4^{th}) order statistics and the global structure of $\mathsf{O}(n;\mathbb{R})$. Its remarkable



(a) Convergence plots of the MSP Algorithm 1 for the deterministic case, with the same initialization.

(b) Average normalized error of MSP Algorithm 2 among 20 trails, varying k and p, with $n = 10$ fixed.

Fig. 5: Use different ℓ^{2k} norms for Algorithm 1 and Algorithm 2.

efficiency (in terms of sample size and computational complexity) as well as its wide range of success suggests the problem merits more refined theoretical analysis in the future. We would very much like to explore if similar analysis and algorithms can be extended to solving the cases of learning overcomplete dictionaries. We would also like to make the learning algorithm robust to measurement noise and outliers.

References

[1] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[2] Yu Bai, Qijia Jiang, and Ju Sun. Subgradient descent learns orthogonal dictionaries. *arXiv preprint arXiv:1810.10702*, 2018.

[3] Chenglong Bao, Jian-Feng Cai, and Hui Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3384–3391, 2013.

[4] Boaz Barak, Jonathan Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *STOC*, 2015.

[5] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[6] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[7] Aapo Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3917–3920, 1997.

[8] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[9] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[11] Saiprasad Ravishankar and Yoram Bresler. Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging. *SIAM Journal on Imaging Sciences*, 8(4):2519–2557, 2015.

[12] Saiprasad Ravishankar and Yoram Bresler. ℓ_0 sparsifying transform learning with efficient optimal updates and convergence guarantees. *IEEE Transactions on Signal Processing*, 9(63):2389–2404, 2015.

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

[14] Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pages 37–1, 2012.

[15] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere. *arXiv preprint arXiv:1504.06785*, 2015.

[16] Jennifer Seberry Wallis. On the existence of Hadamard matrices. *Journal of Combinatorial Theory, Series A*, 21(2):188–195, 1976.

[17] Eric W Weisstein. Bézout's theorem. 1999.