A Fast Holistic Algorithm for Complete Dictionary Learning via $\ell^4$ Norm Maximization

Yuexiong 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 $\ell^1$ norm to exploit sparsity and learns the dictionary one column at a time, we propose instead to maximize $\ell^4$ 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 $\ell^1$-based methods. Through extensive experiments, we also show that, somewhat remarkably, maximizing $\ell^4$ 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 $y \in \mathbb{R}^n$ is assumed to be a sparse superposition of columns of a complete dictionary $D \in \mathbb{R}^{n \times n}$: $y = D_x$, where $x \in \mathbb{R}^n$ is a sparse (coefficient) vector. A typical statistical model for the sparse coefficient is that entries of $x$ are iid Bernoulli Gaussian $\{x_i\} \sim_{iid} \mathcal{BG}(\theta)$ [3], [13], [16].

Suppose we are given a collection of sample signals $Y = [y_1, y_2, \ldots, y_p] \in \mathbb{R}^{n \times p}$, each of which is generated as $y_i = D_x$. Write $X_o = [x_1, x_2, \ldots, x_p] \in \mathbb{R}^{n \times p}$. In this notation, $Y = D_o X_o$.

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

Because $Y$ is sparsely generated, the optimal estimate $D_o$ should make the associated coefficients $X_o$ maximally sparse. In other words, $\ell^0$-norm of $X_o$ should be as small as possible:

$$\min_{X,D} \|X\|_0, \text{ subject to } Y = DX, \ D \in \mathbb{O}(n; \mathbb{R}). \quad (1.2)$$

Under fairly mild conditions, globally minimizing the $\ell^0$ norm recovers the true dictionary $D_o$. But such global minimization of the $\ell^0$ norm is challenging. Typically, as in the K-SVD algorithm [2], [14], one resorts to local heuristics such as orthogonal matching pursuit. This approach has been widely practiced but is challenging to give guarantees. We will compare these algorithms with ours.

1) **Methods based on minimizing $\ell^1$ norm:** Alternatively, a number of works [3], [11], [12], [15], [16] have considered the $\ell^1$ norm as a convex and continuous relaxation of $\ell^0$ and solved variants of the following problem instead:

$$\min_{X,D} \|X\|_1, \text{ subject to } Y = DX, \ D \in \mathbb{O}(n; \mathbb{R}). \quad (1.3)$$

Although $\ell^1$-minimization has been widely practiced in dictionary learning, rigorous justification for its global optimality and correctness was only recently given in [16]. That work is based on the observation that, for a complete dictionary learning $Y = DX$, rows of $Y$ and $X$ span the same subspace: row (X) = row (Y). Hence, if $d$ is a column of $D$, then $d^*Y$ would correspond to a row of $X$, therefore highly sparse. Under certain conditions, one can correctly recover each of the $n$ columns of $D$, by minimizing the $\ell^1$ norm of $d^*Y$ over a sphere:

$$\min_{d \in \mathbb{R}^n} \|d^*Y\|_1, \text{ subject to } \|d\|_2^2 = 1. \quad (1.4)$$

Although [16] provides theoretical guarantees for the complete dictionary learning problem, it requires to solve $n$ optimization programs of the kind (1.4) to find all $n$ columns $d_i$ of the desired dictionary $D$.

2) **Methods based on higher order norms or statistics:** The 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 $\mathbb{O}(n; \mathbb{R})$. An observation comes from the fact that over the sphere $S^{n-1} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$:

$$\arg \max_{x \in S^{n-1}} \|x\|_{1} = \arg \min_{x \in S^{n-1}} \|x\|_0. \quad (1.5)$$

That is, global maxima of the $\ell^1$ norm over the sphere are the same as minima of $\ell^0$. Therefore, instead of using $\ell^1$ norm in (1.4), we could promote the sparsity of $d^*Y$ by considering the following program:

$$\max_{d \in \mathbb{R}^n} \|d^*Y\|_{4}, \text{ subject to } \|d\|_2^2 = 1. \quad (1.6)$$

Unlike $\ell^0$ or $\ell^1$, the $\ell^4$ norm is smooth so there is no reason to solve (1.6) $n$ times separately. We can directly maximize the sum

---

1The number of non-zero entries

2In this paper, we use $d^*$ or $D^*$ to denote (conjugate) transpose.
of $\ell^4$ norm\(^3\) of all rows of $D^* Y$ altogether while enforcing the orthogonality constraint on $D$:

$$
\max_D \|D^* Y\|^4_4, \quad \text{subject to} \quad D \in O(n; \mathbb{R}). \tag{1.7}
$$

We should note that $4^{th}$ order statistical cumulant has been widely used in blind source separation or independent component analysis (ICA) since the 90’s, see [9], [10] and references therein. So if $x$ are independent components, by finding extrema of the so-called kurtosis: $\text{kur}(d^4 y) \doteq \mathbb{E}[d^4 y] - 3 \mathbb{E}[(d^2 y)^2]^2$, one can identify one independent (non-Gaussian) component $x_i$ at a time. Algorithm wise, this is similar to using the $\ell^4$ minimization [4] to identify one column $d_i$ at a time for $D$. Fast fixed-point like algorithms have been developed for this purpose [9], [10]. If $x$ are indeed i.i.d. Bernoulli Gaussian, with $\|d\|^2_2 = 1$, the second term in kurt$(d^4 y)$ would become a constant. The objective of ICA coincides with maximizing the sparsity-promoting $\ell^4$ norm over a sphere [15].

The use of $\ell^4$ norm can also be justified from the perspective of sum of squares (SOS). The work of [5] shows that in theory, when $x$ is sufficiently sparse, one can utilize properties of higher order sum of squares polynomials (such as the fourth order polynomials) to correctly recover $D$, again one column $d_i$ at a time.

In this work, we show that all the columns $d_i$ of $D$ should be learned together by solving the program (1.7) in a holistic fashion so that additional orthogonality constraints among all columns $d_i d_j = \delta_{ij}$ can be exploited together. 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 [5] or $\ell^4$ minimization [16], at least for the complete case.

**Remark 1.1 (Maximizing $\ell^{2k}$ norm):** Conceptually, to promote sparsity, one could also consider maximizing $\ell^{2k}$ norm\(^4\) of $D^* Y$ for any $k \geq 2$. Most analysis and results given in this paper would extend to these general cases. Nevertheless, as we will discuss in Section IV, 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 [11], there are some intrinsic ambiguities regarding recovering $D_o$ in a holistic fashion: given any signed permutation matrix $P \in \mathbb{SP}(n)$\(^5\) we have:

$$
Y = D_o X_o = D_o PP^* X_o,
$$

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

**C. Main Results and Contributions**

1) **Statistical justification:** Suppose that a signal matrix $Y \in \mathbb{R}^{n \times p}$ is randomly generated from [11], we claim that the expected behaviors of solving the following $\ell^4$ norm maximization problem:

$$
\max_A \|AY\|^4_4, \quad \text{subject to} \quad A \in O(n; \mathbb{R}), \tag{1.8}
$$

are largely characterized by the following (deterministic) program:

$$
\max_A \|AD_o\|^4_4, \quad \text{subject to} \quad A \in O(n; \mathbb{R}), \tag{1.9}
$$

whose global optima are $D_o^*$ up to arbitrary signed permutations (that is, $A_o$ shall satisfy $A_o D_o \in \mathbb{SP}(n)$). 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 at a time, we introduce a novel matching, stretching, and projection (MSP) algorithm that solves the programs (1.8) and (1.9) directly for the entire $D \in O(n; \mathbb{R})$. The algorithm exploits the statistics of $\ell^4$ and global geometry of $O(n; \mathbb{R})$ 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 (1.8), which goes well beyond any existing theoretical guarantees [5], [16] for the complete dictionary case.

**D. Notations**

We use a bold uppercase and lowercase letters to denote matrices and vectors respectively: $X \in \mathbb{R}^{n \times p}$, $x \in \mathbb{R}^n$. We reserve small letter for scalar: $x \in \mathbb{R}$. We use $\|X\|_F$ to denote the element-wise $\ell^2$ norm of matrix $X$. We use $D_o$ to denote the ground truth dictionary, and $A$ is an estimate for $D_o$ from solving (1.8). We use $\circ$ to denote the Hadamard product: $\forall A, B \in \mathbb{R}^{n \times m}$, $\{A \circ B\}_{i,j} = a_{i,j} b_{i,j}$, and $\{A^\circ\}_i = \{a_i^\circ\} = a_i^\circ$. The $\ell^4$ norm of matrix $A \in \mathbb{R}^{n \times p}$ is $\|A\|_4 = \sum_{i,j} a_{i,j}^4$. We use $\mathbb{E}[X]$, $\mathbb{E}(X)$ to denote the expectation of $X$. We use $\mathbb{E} \mathbb{E}(X) = \mathbb{E}(X^\circ)$ to denote the Hadamard product.

Given an input data matrix $Y$ randomly generated from (1.1), for any orthogonal matrix $A \in O(n; \mathbb{R})$, we define $f : O(n; \mathbb{R}) \times \mathbb{R}^{n \times p} \mapsto \mathbb{R}$ as the $4^{th}$ power of $\ell^4$ norm of $AY$:

$$
\hat{f}(A, Y) = \mathbb{E}_{X_o}[\|AX_o\|^4_4]. \tag{1.10}
$$

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

$$
f(A) = \mathbb{E}_{X_o} [\|AX_o\|^4_4] = \mathbb{E}_{X_o} [\|AY\|^4_4]. \tag{1.11}
$$

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

**E. Organization of the Paper**

Rest of the paper is organized as follows. In Section II we characterize the global maximizers of (1.8) 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 the proposed 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

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

to recover the ground truth dictionary $D_o$.

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

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

$$
\mathbb{P} \left( \left| \frac{\hat{f}(A, Y)}{np} - \frac{f(A)}{np} \right| > \varepsilon \right) \leq O \left( \frac{n^4 \theta^3}{p \varepsilon^2} \right). 
$$

(II.2)

This indicates that the (random) function $\hat{f}(A, Y)$ behaves like its expectation $f(A)$ 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 $1/\varepsilon$, or more precisely $p > O(n^4/\varepsilon^2)$.

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

$$
\max_A f(A) = \mathbb{E}_{X_s} \left[ \| AY \|_2^4 \right], \quad \text{subject to } A \in O(n; \mathbb{R}).
$$

(II.3)

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

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

- $\frac{1}{np} f(A) = (1 - \theta) g(AD_o) + \theta n$.
- $\frac{1}{np} f(A) \leq n$, with equality holds if and only if $AD_o \in \text{SP}(n)$.

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

$$
A = \arg \max_A f(A) \quad \text{if and only if} \quad A = \arg \max_{A \in \text{SP}(n)} g(AD_o).
$$

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

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

(II.4)

A. Related Optimization Methods

Although (II.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 $\mathcal{L}(A, \lambda) = -\|AY\|_2^4 + (AA - I, A)$ will go to negative infinity due to the concavity of the objective function $\|AY\|_2^4$. 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 $O(n; \mathbb{R})$.

Another natural way to optimize (II.8) is to apply Riemannian gradient (or projected gradient) type methods on the group $O(n; \mathbb{R})$. 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 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. $\ell^4$ Maximization over $O(n; \mathbb{R})$ via an MSP Algorithm

We now introduce the matching, stretching and projection (MSP) algorithm to solve problems (II.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 (II.8) concentrates on the $\ell^4$ norm maximization problem (II.4) w.h.p., we first introduce the MSP algorithm for solving (II.4):

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

**Algorithm 1** MSP for $\ell^4$ Maximization over Orthogonal Group

1: Given any $D_o \in O(n; \mathbb{R})$. \> Ground truth $D_o$
2: Initialize: $A_0 \in O(n; \mathbb{R})$. \> Initialize $A_0$ for iteration
3: for $t = 0, 1, \ldots$ do
4: \quad $\partial A_t = 4(A_tD_o)\odot D_0^*$. \> $\nabla A \| AD_o \|_4^4 = 4(AD_o)\odot D_0^*$
5: \quad $\nabla V^\ast \leftarrow \text{svd}(\partial A_t)$. \> Project $A$ onto orthogonal group
6: \quad $A_{t+1} = UV^\ast$. \> $A_{t+1}$
7: end for
8: Output: $A_{t+1} \| A_{t+1}D_o \|_4^4 / n$.

Note that in the output we normalize $\| AD_o \|_4^4$ by dividing $n$, because the global maximum of $\| AD_o \|_4^4$ is $n$ and the output is therefore normalized to 1. In Step 4 of the MSP algorithm, the calculation of $\partial A_t = 4(A_tD_o)\odot D_0^*$ does not require knowledge of $D_o$. It is merely the gradient of the objective function

$$
\nabla A g(AD_o) = \nabla A \| AD_o \|_4^4 = 4(AD_o)\odot D_0^*.
$$

However, one shall not mistake the MSP algorithm as a gradient descent type algorithm. In fact, in Step 4, the scale of $\partial A_t$ is very large and the iterates are not incremental local updates. Due to the scale invariant of SVT\(^1\) one can even scale $\partial A_t$ 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 $A_t$ with the true $D_o$. Then the elementwise cubic function $(\cdot)^{3}$ stretches all entries of $A_tD_o$ by promoting

\(^1\)The projection of a square matrix onto the orthogonal group is scale invariant: $\forall \alpha > 0$, the projection of $\alpha A$ and $\alpha A$ onto $O(n; \mathbb{R})$ are the same.
the large ones and suppressing the small ones. \(\partial A_t\) is the correlation between so “sparsified” pattern and the original basis \(D_o\), which is then projected back onto the closest orthogonal matrix \(A_{t+1}\) in Frobenius norm.

Repeating this “matching, stretching, and projection” process, \(A_t D_o\) is increasingly sparsified while ensuring the orthogonality of \(A_t\). Ideally the process will stop when \(A_t D_o\) 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 \[10\], later analysis will show that it achieves super-linear convergence.

b) The Random Case: For the original dictionary learning problem \[8\]:

\[
\max_A \hat{f}(A, Y) = \|AY\|_4, \quad \text{subject to} \quad A \in O(n; \mathbb{R}),
\]

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

**Algorithm 2 MSP for \(\ell^4\) Maximization Based Dictionary Learning**

1. Given: \(Y = D_o X_o \in \mathbb{R}^{n \times p}\), \(D_o \sim O(n; \mathbb{R})\).
2. Initialize: \(A_0 \in O(n; \mathbb{R})\).
3. For \(t = 0, 1, \ldots\) do
4. \(\partial A_t = 4(4(A Y)^3 Y^* + \hat{f}(A, Y))\).
5. \(\nabla A \|AY\|_4^4 = 4(AY)^3 Y^*\).
6. \(A_{t+1} = UV^*\).
7. End for
8. Output: \(A_{t+1}, \|A_{t+1} Y\|_4^4 / 3np\theta, \|A_{t+1} D_o\|_4^4 / n\).

Note that in the output we also normalize \(\|AY\|_4^4\) 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. In Step 4, the estimate \(A_t\) is first matched with the observation \(Y\). The cubic function \(\|\cdot\|^4\) re-scales the results and promotes entry-wise sparsity of \(X_o = A_t Y\) accordingly. Again, here \(\partial A_t\) is the gradient \(\nabla A \hat{f}(A, Y)\) of the objective function, but because of its large scale, the algorithm is not performing gradient descent.

Although the algorithm and the objective function are random here, Lemma 3.3 below shows that \(\nabla A \hat{f}(A, Y)\) concentrates on its expectation when \(p\) increases. Theorem 3.5 further shows its linear relationship with \(\nabla A g(D_o)\).

**Lemma 3.1 (Concentration Bound of \(\nabla A \hat{f}(A, Y)\))**:

\[\forall A \in O(n; \mathbb{R}), \forall \varepsilon > 0, \nabla A \hat{f}(A, Y) \text{ concentrates to its expectation with the following bound}
\]

\[\mathbb{P}\left(\left|\frac{1}{p} \nabla A \hat{f}(A, Y) - \frac{1}{p} \mathbb{E}_{X_o} [\nabla A \hat{f}(A, Y)]\right|^2 > \varepsilon\right) \leq O\left(\frac{n^2 \theta^4}{p^2}\right).
\]

**Theorem 3.2 (Expectation of \(\nabla A \hat{f}(A, Y)\))**: With \(Y\) defined as in \[4\], the expectation of \(\nabla A \hat{f}(A, Y)\) satisfies:

\[\mathbb{E}_{X_o} \nabla A \hat{f}(A, Y) = \text{3p}(1 - \theta) \nabla A g(D_o) + 12p\theta^2 A. \quad (\text{III.1})\]

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

With these results, the direction of \(\nabla A \hat{f}(A, Y)\) is a linear combination of \(\nabla A g(D_o)\) and \(A\). So we expect the stretching \(\partial A_t = 4(A Y)^3 Y^*\) in Step 4 of Algorithm 2 also promotes the sparsity of \(A_s D_o\) w.h.p., as long as \(\Theta \in (0, 1)\). Moreover, the stretching direction of \(\nabla A \hat{f}(A, Y)\) approximates \(\nabla A g(D_o)\) better with smaller \(\Theta\) (sparser \(X_o\)), which suggests that the learning algorithm is more likely to succeed with sparser \(X_o\), 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 \(O(n; \mathbb{R})\). So without loss of generality, we only need to provide convergence analysis for the case \(D_o = I\).

When \(D_o = I\), we want to show the MSP algorithm converges to a signed permutation matrix for the optimization problem:

\[\max g(A) = \|A\|_4^4, \quad \text{subject to} \quad A \in O(n; \mathbb{R}),\]

starting from any initial \(A_0\) on \(O(n; \mathbb{R})\). One can easily show that all critical points \(W \in \mathbb{R}^{n \times n}\) of \(\ell^4\)-norm over the orthogonal group satisfy the following condition:

\[(W^{3})^* W = W^{'} W^{3}, \quad W^{'} W = I. \quad (\text{III.2})\]

Since the orthogonal group \(O(n; \mathbb{R})\) is a continuous manifold in \(\mathbb{R}^{n \times n}\) \[1\], \[7\], this indicates that critical points of \(g(W) = \|W\|_4^4\) over the orthogonal group \(W \in O(n; \mathbb{R})\) have measure 0.

**Lemma 3.3 (Discreteness of Critical Points)**: All global maximizers of \(\ell^4\)-norm over the orthogonal group are isolated (nondegenerate) critical points.

For now we have the following relation between fixed points of the MSP Algorithm \[1\] when \(D_o = I\) and the critical points of \(g(\cdot)\).

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

Although the function \(g(W) = \|W\|_4^4\) may have many critical points, the signed permutation group \(SP(n)\) are the only global maximizers. As recent work has shown \[16\], such discrete symmetry helps regulate the global landscape of the objective function and makes it amenable to global optimization. Indeed, we have observed through 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 \[1\]. That is, when the initial orthogonal matrix \(A\) 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 \(I\).

**Theorem 3.5 (Local Convergence Rate of the MSP Algorithm)**:

Given \(A \in O(n; \mathbb{R})\), if \(\|A - I\|_p^p < \varepsilon\), and let \(A'\) denote the output of the MSP Algorithm \[1\] after one iteration: \(A' = U V^*, \) where \(U V^* = \text{svd}(A^3)\), then \(\|A' - I\|_p^p \leq O(\varepsilon^2)\).

**Theorem 3.6** 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 3.5 (Maximizing \(\ell^2k\) norm)**: One can easily extend the MSP algorithm to maximize \(\ell^2k\) 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 \(\left(\frac{\|\cdot\|_{\ell^2k}}{\|\cdot\|_{\ell^4}}\right)^{2k-1}\)
sparsifies the matrix more significantly with a larger $k$.

See Section [IV-E] for experimental verification. However, as we discussed in the Section [IV] 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 the proposed algorithm with the classic heuristic KSVD algorithm [2] and the latest provably correct dictionary learning method [3] based on minimizing $\ell^1$ 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 [2] and the latest subgradient method [3] for different choices of $n, p$ under the same sparsity level $\theta = 0.3$. As shown in table [I] the MSP algorithm is significantly faster than both algorithms in all trials. Further more, the MSP algorithm has the potential for large scale experiments: it only takes 374.2 seconds to learn a $400 \times 400$ dictionaries from 160,000 samples. While the previous algorithms either fail to find the correct dictionary or barely applicable. Within statistical errors, the MSP algorithm gives slightly smaller values for $\|AD_o\|_4^4/n$ in some trials. But the subgradient method [3] uses information of the ground truth dictionary $D_o$ in their stopping criteria. The MSP algorithm removes this dependency with only mild loss in accuracy.

<table>
<thead>
<tr>
<th>Trials</th>
<th>Error</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>12.35%</td>
<td>51.2s</td>
</tr>
<tr>
<td>(b)</td>
<td>8.63%</td>
<td>244.4s</td>
</tr>
<tr>
<td>(c)</td>
<td>6.15%</td>
<td>684.9s</td>
</tr>
<tr>
<td>(d)</td>
<td>6.16%</td>
<td>1042.3s</td>
</tr>
<tr>
<td>(e)</td>
<td>13.07%</td>
<td>5401.9s</td>
</tr>
</tbody>
</table>

Table I: Comparison experiments with [2], [3] in different trials 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) $n = 200, p = 4 \times 10^5, \theta = 0.3$; (e) $n = 400, p = 16 \times 10^4, \theta = 0.3$. $Y$ is generated from [17]. Recovery error is measured as \(1 - \|AD_o\|_4^4/n\), since Lemma [2.3] shows that a perfect recovery gives $\|AD_o\|_4^4/n = 1$. 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 [I]a presents one trial of the proposed MSP Algorithm [2] for dictionary learning with $\theta = 0.3, n = 100, p = 40,000$. The result corroborates with statements in Lemma [2.1] and Lemma [2.2] maximizing $f(A, Y)$ is largely equivalent to optimizing $g(AD_o)$, and both values reach global maximum at the same time. Meanwhile, this result also shows the MSP algorithm is able to find the global maximum at ease, since $g(AD_o)$ reaches its maximal value 1 (with minor errors) by maximizing $f(A, Y)$. In Figure [I]b, we test the MSP Algorithm [2] in higher dimension $n = 400, p = 1.6 \times 10^5, \theta = 0.3$. In both cases, the proposed algorithm is surprisingly efficient: it only takes around 20 iterations to recover a 100-dimensional dictionary and 50 iterations for a 400-dimensional dictionary.

\[\text{In fact, one can show that if } 2k \to \infty, \text{ the corresponding MSP algorithm converges with only one iteration for the deterministic case!}\]
order statistics and the global structure of $O(\nu; \mathbb{R})$. Its remarkable 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**


