DICTIONARY LEARNING FOR SPARSE REPRESENTATION: COMPLEXITY AND ALGORITHMS

Meisam Razaviyayn *Hung-Wei Tseng* *Zhi-Quan Luo*

Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455

ABSTRACT

In this paper we consider the dictionary learning problem for sparse representation. We first show that this problem is NP-hard and then propose an efficient dictionary learning scheme to solve several practical formulations of this problem. Unlike many existing algorithms in the literature, such as K-SVD, our proposed dictionary learning scheme is theoretically guaranteed to converge to the set of stationary points under certain mild assumptions. For the image denoising application, the performance and the efficiency of the proposed dictionary learning scheme are comparable to that of K-SVD algorithm in simulation.

Index Terms— Dictionary learning, sparse representation, computational complexity, K-SVD.

1. INTRODUCTION

The idea of representing a signal with few samples/observations dates back to the classical result of Kotelnikon, Nyquist, Shannon, and Whittaker [1–5]. This idea has evolved over time, and culminated to the *compressive sensing* concept in recent years [6, 7]. The *compressive sensing* or *sparse recovery* approach relies on the observation that many practical signals can be sparsely approximated in a suitable over-complete basis (i.e., a dictionary). In other words, the signal can be approximately written as a linear combination of only a few components (or *atoms*) of the dictionary. This observation is a key to many lossy compression methods such as JPEG and MP3.

Theoretically, the exact sparse recovery is possible with high probability under certain conditions. More precisely, it is demonstrated that if the linear measurement matrix satisfies some conditions such as null space property (NSP) or restricted isometry property (RIP), then the exact recovery is possible [6, 7]. These conditions are satisfied with high probability for different matrices such as Gaussian random matrices, Bernoulli random matrices, and partial random Fourier matrices.

In addition to the theoretical advances, compressive sensing has shown great potential in various applications. For example, in the nuclear magnetic resonance (NMR) imaging application, compressive sensing can help reduce the radiation time [8, 9]. Moreover, the compressive sensing technique has been successfully applied to many other practical scenarios including sub-Nyquist sampling [10, 11], compressive imaging [12, 13], and compressive sensor networks [14, 15], to name just a few.

In some of the aforementioned applications, the sensing matrix and dictionary are pre-defined using application domain knowledge. However, in most applications, the dictionary is not known a-priori and must be learned using a set of training signals. It has been observed that learning a good dictionary can substantially improve the compressive sensing performance, see [16–22]. In these applications, dictionary learning is the most crucial step affecting the performance of the compressive sensing approach.

To determine a high quality dictionary, various learning algorithms have been proposed; see, e.g., [16, 22–24]. These algorithms are typically composed of two major steps: 1) finding an approximate sparse representation of the training signals 2) updating the dictionary using the sparse representation.

In this paper, we consider the dictionary learning problem for sparse representation. We first establish the NP-hardness of this problem. Then we consider different formulations of the dictionary learning problem and propose several efficient algorithms to solve this problem. In contrast to the existing dictionary training algorithms [16, 22, 23], our methods neither solve Lasso-type subproblems nor find the active support of the sparse representation vector at each step; instead, they require only simple inexact updates in closed form. Furthermore, unlike most of the existing methods in the literature, e.g., [16,22], the iterates generated by the proposed dictionary learning algorithms are theoretically guaranteed to converge to the set of stationary points under certain mild assumptions.

2. PROBLEM STATEMENT

Given a set of training signals $\mathrm{Y} = \{\mathbf{y}_i \in \mathbb{R}^n \mid i = 1, 2, \ldots, N\}$, our task is to find a dictionary $\mathrm{A} = \{\mathbf{a}_i \in \mathbb{R}^n \mid i = 1, 2, \ldots, k\}$that can sparsely represent the training signals in the set Y. Let $\mathbf{x}_i \in \mathbb{R}^k,\ i = 1, \ldots, N$, denote the coefficients of sparse representation of the signal

This research is supported in part by the National Science Foundation, grant number DMS-1015346.

$\mathbf{y}_i$, i.e., $\mathbf{y}_i = \sum_{j=1}^{k} \mathbf{a}_j x_{ij}$, where x_{ij} is the j-th component of signal $\mathbf{x}_i$. By concatenating all the training signals, the dictionary elements, and the coefficients, we can define the matrices $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N]$, $\mathbf{A} \triangleq [\mathbf{a}_1, \ldots, \mathbf{a}_k]$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$. Having these definitions in our hands, the dictionary learning problem for sparse representation can be stated as

$$\min_{\mathbf{A},\mathbf{X}} \ \mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) \quad \text{s.t. } \mathbf{A} \in \mathcal{A}, \ \mathbf{X} \in \mathcal{X}, \tag{1}$$

where $\mathcal{A}$ and $\mathcal{X}$ are two constraint sets. The function $\mathrm{d}(\cdot,\cdot,\cdot)$ measures our model goodness of fit. In the next section, we analyze the computational complexity of one of the most popular forms of problem (1).

3. COMPLEXITY ANALYSIS

Consider a special case of problem (1) by choosing the distance function to be the Frobenius norm and imposing sparsity by considering the constraint set $\mathcal{X} = \{\mathbf{X} \in \mathbb{R}^{k\times N} \ \big| \ \|\mathbf{x}_i\|_0 \leq s\}$. Then the optimization problem (1) can be re-written as

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2, \ \text{s.t. } \|\mathbf{x}_i\|_0 \leq s, \ \forall\, i = 1, \ldots, N. \tag{2}$$

This formulation is very popular and is considered in different studies; see, e.g., [22, 25]. The following theorem characterizes the computational complexity of (2) by showing its NP-hardness. In particular, we show that even for the simple case of $s = 1$ and $k = 2$, problem (2) is NP-hard. To state our result, let us define the following concept: let $(\mathbf{A}^*, \mathbf{X}^*)$ be a solution of (2). For $\epsilon > 0$, we say a point $(\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$ is an *ϵ-optimal solution* of (2) if $\|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}\|_F^2 \leq \|\mathbf{Y} - \mathbf{A}^*\mathbf{X}^*\|_F^2 + \epsilon$.

Theorem 1 *Assume $s = 1$ and $k = 2$. Then finding an ϵ-optimal algorithm for solving* (2) *is NP-hard. In other words, there is no polynomial time algorithm in $N, n, \log\lceil \frac{1}{\epsilon} \rceil$ that can solve* (2) *to ϵ-optimality, unless $P = NP$.*

The proof of Theorem (1) is lengthy and will not be presented here due to space limitation.

It is worth noting that the above NP-hardness result is different from (and is not a consequence of) the compressive sensing NP-hardness result in [26]. In fact, for a fixed sparsity level s, the compressive sensing problem is no longer NP-hard, while the dictionary learning problem considered herein remains NP-hard (see Theorem 1).

4. ALGORITHMS

4.1. Optimizing the goodness of fit

In this section, we assume that the function $\mathrm{d}(\cdot)$ is composed of a smooth part and a non-smooth part for promoting sparsity, i.e., $\mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) = \mathrm{d}_1(\mathbf{Y}, \mathbf{A}, \mathbf{X}) + \mathrm{d}_2(\mathbf{X})$, where d_1 is smooth and d_2 is continuous and possibly non-smooth. Let us further assume that the sets $\mathcal{A}, \mathcal{X}$ are closed and convex. Our approach to solve (1) is to apply the general block successive upper-bound minimization framework developed in [27]. More specifically, we propose to alternately update the variables $\mathbf{A}$ and $\mathbf{X}$. Let $(\mathbf{A}^r, \mathbf{X}^r)$ be the point obtained by the algorithm at iteration r. Then, we select one of the following methods to update the dictionary variable $\mathbf{A}$ at iteration $r+1$:

(a) $\mathbf{A}^{r+1} \leftarrow \arg\min_{\mathbf{A}\in\mathcal{A}} \mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}^r)$

(b) $\mathbf{A}^{r+1} \leftarrow \arg\min_{\mathbf{A}\in\mathcal{A}} \ \langle \nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r), \mathbf{A}\rangle + \frac{\tau_a^r}{2}\|\mathbf{A} - \mathbf{A}^r\|_F^2 = \mathcal{P}_{\mathcal{A}}\left(\mathbf{A}^r - \frac{1}{\tau_a^r}\nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r)\right)$

and we update the variable $\mathbf{X}$ by

- $\mathbf{X}^{r+1} \leftarrow \arg\min_{\mathbf{X}\in\mathcal{X}} \ \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r), \mathbf{X}\rangle + \frac{\tau_x^r}{2}\|\mathbf{X} - \mathbf{X}^r\|_F^2 + \mathrm{d}_2(\mathbf{X})$.

Here the operator $\langle\cdot,\cdot\rangle$ denotes the inner product; the superscript r represents the iteration number; the notation $\mathcal{P}_{\mathcal{A}}(\cdot)$ is the projection operator to the convex set $\mathcal{A}$; and the constants $\tau_a^r \triangleq \tau_a(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r)$ and $\tau_x^r \triangleq \tau_x(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r)$ are chosen such that

$$\mathrm{d}_1(\mathbf{Y}, \mathbf{A}, \mathbf{X}^r) \leq \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r) + \langle \nabla_{\mathbf{A}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^r, \mathbf{X}^r), \mathbf{A} - \mathbf{A}^r\rangle + \frac{\tau_a^r}{2}\|\mathbf{A} - \mathbf{A}^r\|_F^2, \ \forall\, \mathbf{A} \in \mathcal{A}$$

and

$$\mathrm{d}(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}) \leq \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r) + \mathrm{d}_2(\mathbf{X}) + \frac{\tau_x^r}{2}\|\mathbf{X} - \mathbf{X}^r\|_F^2 + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y}, \mathbf{A}^{r+1}, \mathbf{X}^r), \mathbf{X} - \mathbf{X}^r\rangle, \ \forall\, \mathbf{X} \in \mathcal{X}. \tag{3}$$

It should be noted that each step of the algorithm requires solving an optimization problem. For the commonly used objective functions and constraint sets, the solution to these optimization problems is often in closed form. In addition, the update rule (b) is the classical gradient projection step which can be viewed as an approximate version of (a). As we will see later, for some special choices of the function $\mathrm{d}(\cdot)$ and the set $\mathcal{A}$, using (b) leads to a closed form update rule, while (a) does not. In the sequel, we specialize this framework to different popular choices of the objective functions and the constraint sets.

Case I: Constraining the total dictionary norm
For any $\beta > 0$, we consider the following optimization problem

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{A}\|_F^2 \leq \beta, \tag{4}$$

where λ denotes the regularization parameter. By simple calculations, we can check that all the steps of the proposed algorithm can be done in closed form. More specifically, using the dictionary update rule (a) will lead to Algorithm 1. In

Algorithm 1 The proposed algorithm for solving (4)

initialize $\mathbf{A}$ randomly such that $\|\mathbf{A}\|_F^2 \leq \beta$
repeat
$\tau_a \leftarrow \sigma_{\max}^2(\mathbf{X})$
$\mathbf{X} \leftarrow \mathbf{X} - \mathcal{S}_{\frac{\lambda}{\tau_a}}(\mathbf{X} - \frac{1}{\tau_a}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}))$
$\mathbf{A} \leftarrow \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \theta\mathbf{I})^{-1}$
until some convergence criterion is met

this algorithm, $\sigma_{\max}(\cdot)$ denotes the maximum singular value; $\theta \geq 0$ is the Lagrange multiplier of the constraint $\|\mathbf{A}\|_F^2 \leq \beta$ which can be found using one dimensional search algorithms such as bisection or Newton. The notation $\mathcal{S}(\cdot)$ denotes the component-wise soft shrinkage operator, i.e., $\mathbf{B} = \mathcal{S}_\gamma(\mathbf{C})$ if

$$\mathbf{B}_{ij} = \begin{cases} \mathbf{C}_{ij} - \gamma & \text{if } \mathbf{C}_{ij} > \gamma \\ 0 & \text{if } -\gamma \leq \mathbf{C}_{ij} \leq \gamma \\ \mathbf{C}_{ij} + \gamma & \text{if } \mathbf{C}_{ij} < -\gamma \end{cases}$$

where $\mathbf{B}_{ij}$ and $\mathbf{C}_{ij}$ denote the (i,j)-th component of the matrices $\mathbf{B}$ and $\mathbf{C}$, respectively.

Case II: Constraining the norm of each dictionary atom
In many applications, it is of interest to constrain the norm of each dictionary atom, i.e., the dictionary is learned by solving:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i \quad (5)$$

In this case, the dictionary update rule (a) cannot be expressed in closed form; as an alternative, we can use the update rule (b), which is in closed form, in place of (a). This gives Algorithm 2. In this algorithm, the set $\mathcal{A}$ is defined as

Algorithm 2 The proposed algorithm for solving (5) and (6)

For solving (5): initialize $\mathbf{A}$ randomly s.t. $\|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i$
For solving (6): initialize $\|\mathbf{A}\|_F^2 \leq \beta$ and $\mathbf{A} \geq 0$
repeat
$\tau_x \leftarrow \sigma_{\max}^2(\mathbf{A})$
For solving (5): $\mathbf{X} \leftarrow \mathbf{X} - \mathcal{S}_{\frac{\lambda}{\tau_x}}(\mathbf{X} - \frac{1}{\tau_x}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}))$
For solving (6): $\mathbf{X} \leftarrow \mathcal{P}_{\mathcal{X}}\left(\mathbf{X} - \frac{1}{\tau_x}\mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}) - \lambda\right)$
$\tau_a \leftarrow \sigma_{\max}^2(\mathbf{X})$
$\mathbf{A} \leftarrow \mathcal{P}_{\mathcal{A}}\left(\mathbf{A} - \frac{1}{\tau_a}(\mathbf{A}\mathbf{X} - \mathbf{Y})\mathbf{X}^T\right)$
until some convergence criterion is met

$$\mathcal{A} \triangleq \{\mathbf{A} \mid \|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i\}$$

Case III: Non-negative dictionary learning with the total norm constraint
Consider the non-negative dictionary learning problem for sparse representation:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \|\mathbf{A}\|_F^2 \leq \beta, \ \mathbf{A}, \mathbf{X} \geq 0 \quad (6)$$

Utilizing the update rule (b) leads to Algorithm 2. Note that in this case, projections to the sets $\mathcal{X} = \{\mathbf{X} \mid \mathbf{X} \geq 0\}$ and $\mathcal{A} = \{\mathbf{A} \mid \|\mathbf{A}\|_F^2 \leq \beta, \mathbf{A} \geq 0\}$ are simple. In particular, to project to the set $\mathcal{A}$, we just need to first project to the set of nonnegative matrices first and then project to the set $\tilde{\mathcal{A}} = \{\mathbf{A} \mid \|\mathbf{A}\|_F^2 \leq \beta\}$.

It is worth noting that Algorithm 2 can also be applied to the case where $\mathcal{A} = \{\mathbf{A} \mid \mathbf{A} \geq 0, \ \|\mathbf{a}_i\|_F^2 \leq \beta_i, \ \forall\, i\}$, since the projection to the constraint set still remains simple.

Case IV: Sparse non-negative matrix factorization
In some applications, it is desirable to have a sparse non-negative dictionary; see, e.g., [28–30]. In such cases, we can formulate the dictionary learning problem as:

$$\min_{\mathbf{A},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_1 \text{ s.t. } \|\mathbf{a}_i\|_1 \leq \theta, \ \forall\, i, \ \mathbf{A}, \mathbf{X} \geq 0 \quad (7)$$

It can be checked that we can again use the essentially same steps of the algorithm in case III to solve (7). The only required modification is in the projection step since the projection should be onto the set $\mathcal{A} = \{\mathbf{A} \mid \mathbf{A} \geq 0, \|\mathbf{a}_i\|_1 \leq \theta, \ \forall\, i\}$. This step can be performed in a column-wise manner by updating each column $\mathbf{a}_i$ to $[\mathbf{a}_i - \rho_i\mathbf{1}]_+$, where $[\cdot]_+$ denotes the projection to the set of nonnegative matrices and $\rho_i \in \mathbb{R}^+$ is a constant that can be determined via one dimensional bisection. The resulting algorithm is very similar (but not identical) to the one in [28]. However, unlike the algorithm in [28], all of our proposed algorithms are theoretically guaranteed to converge, as shown in Theorem 2.

Theorem 2 *The iterates generated by the algorithms in cases I-IV converge to the set of stationary points of the corresponding optimization problems.*

Proof: Each of the proposed algorithms in cases I-IV is a special case of the block successive upper-bound minimization approach [27]. Therefore, [27, Theorem 2] guarantees the convergence of the proposed methods.

4.2. Constraining the goodness of fit

In some practical applications, the goodness of fit level may be known *a-priori*. In these cases, we may be interested in finding the sparsest representation of the data for a given goodness of fit level. In particular, for a given $\alpha > 0$, we consider

$$\min_{\mathbf{A},\mathbf{X}} \ \|\mathbf{X}\|_1 \quad \text{s.t. } \mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) \leq \alpha, \ \mathbf{A} \in \mathcal{A}, \ \mathbf{X} \in \mathcal{X}. \quad (8)$$

For example, when the noise level is known, the goodness of fit function can be set as $\mathrm{d}(\mathbf{Y}, \mathbf{A}, \mathbf{X}) = \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2$. We propose an efficient method (Algorithm 3) to solve (8), where the constant τ_x is chosen according to criterion in (3).

It is clear that Algorithm 3 is not a special case of block coordinate descent method [31] or even the block successive upper-bound minimization method [27]. Nonetheless, the convergence of Algorithm 3 is guaranteed in light of the following theorem.

Algorithm 3 The proposed algorithm for solving (8)

initialize $\mathbf{A}$ randomly s.t. $\mathbf{A} \in \mathcal{A}$ and find a feasible $\mathbf{X}$
repeat
$\bar{\mathbf{X}} \leftarrow \mathbf{X}$
$\mathbf{X} \leftarrow \arg\min_{\mathbf{X}\in\mathcal{X}} \|\mathbf{X}\|_1 \quad \text{s.t.} \quad \mathrm{d}_1(\mathbf{Y},\mathbf{A},\bar{\mathbf{X}}) + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y},\mathbf{A},\bar{\mathbf{X}}), \mathbf{X}-\bar{\mathbf{X}}\rangle + \frac{\tau_x}{2}\|\mathbf{X}-\bar{\mathbf{X}}\|_F^2 + \mathrm{d}_2(\mathbf{X}) \le \alpha$
$\mathbf{A} \leftarrow \arg\min_{\mathbf{A}\in\mathcal{A}} \mathrm{d}(\mathbf{Y},\mathbf{A},\mathbf{X})$
until some convergence criterion is met

Theorem 3 *Assume that* $(\bar{\mathbf{X}}, \bar{\mathbf{A}})$ *is a limit point of the iterates generated by Algorithm 3. Furthermore, assume that the subproblem for updating* $\mathbf{X}$ *is strictly feasible at* $(\bar{\mathbf{X}}, \bar{\mathbf{A}})$*, i.e., there exists* $\tilde{\mathbf{X}} \in \mathcal{X}$ *such that* $\mathrm{d}_1(\mathbf{Y},\bar{\mathbf{A}},\bar{\mathbf{X}}) + \langle \nabla_{\mathbf{X}} \mathrm{d}_1(\mathbf{Y},\bar{\mathbf{A}},\bar{\mathbf{X}}), \tilde{\mathbf{X}}-\bar{\mathbf{X}}\rangle + \frac{\tau_x}{2}\|\tilde{\mathbf{X}}-\bar{\mathbf{X}}\|_F^2 + \mathrm{d}_2(\tilde{\mathbf{X}}) < \alpha$. *Then* $(\bar{\mathbf{X}}, \bar{\mathbf{A}})$ *is a stationary point of* (8).

This theorem is similar to [32, Property 3]. However, the proof here is different due to the lack of smoothness in the objective function. The proof is omitted due to the space limitation.

5. NUMERICAL EXPERIMENTS

In this section, we apply the proposed sparse dictionary learning method, namely algorithm 2, to the image denoising application; and compare its performance with that of the K-SVD algorithm proposed in [18] (and summarized in Algorithm 4). As a test case, we use the image of Lena corrupted by additive Gaussian noise with various variances (σ^2).

In Algorithm 4, $\mathbf{R}_{i,j}\mathbf{S}$ denotes the image patch centered at (i,j) coordinate. In step 2, dictionary $\mathbf{A}$ is trained to sparsely represent *noisy* image patches by using either K-SVD algorithm or Algorithm 2. The term $\mathbf{x}_{i,j}$ denotes the sparse representation coefficient of the patch (i,j). In K-SVD, it (approximately) solves ℓ_0-norm regularized problem (9) by using orthogonal matching pursuit (OMP) to update $\mathbf{X}$. In our approach, we use Algorithm 2 with $\mathcal{A} = \{\mathbf{A} \mid \|\mathbf{a}_i\| \le 1, \forall\, i = 1, \cdots, N\}$ to solve the ℓ_1-penalized dictionary learning formulation (10). We set $\mu_{i,j} = c(0.0015\sigma + 0.2),\ \forall\, i,j$, in (10) with $c = \frac{1}{I\times J}\sum_{i,j}\|\mathbf{R}_{i,j}S\|_2$, and $I \times J$ denotes the total number of image patches. This choice of the parameter μ_{ij} intuitively means that we emphasize on sparsity more in the presence of stronger noise. Numerical values $(0.0015, 0.2)$ are determined experimentally. The final denoised image $\mathbf{S}$ is obtained by (11) and setting $\beta = 30/\sigma$, as suggested in [18].

σ/PSNR	DCT	K-SVD	Algorithm 2
20/22.11	32	**32.38**	30.88
60/12.57	26.59	**26.86**	26.37
100/8.132	24.42	24.45	**24.46**
140/5.208	22.96	22.93	**23.11**
180/3.025	21.73	21.69	**21.96**

Table 1. Image denoising result comparison on "Lena" for different noise levels. Values are averaged over 10 Monte Carlo simulations.

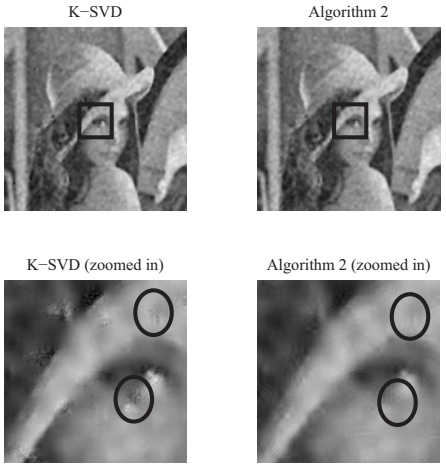

Fig. 1. Sample denoised images ($\sigma = 100$).

Algorithm 4 Image denoising using K-SVD or algorithm 2

Input: noisy image $\mathbf{Y}$, noise variance σ^2
Output: denoised image $\mathbf{S}$
1: Initialization: $\mathbf{S} = \mathbf{Y}$, $\mathbf{A} =$ overcomplete DCT dictionary
2: Dictionary learning:
K-SVD:

$$\min_{\mathbf{A},\mathbf{X}} \sum_{i,j} \mu_{ij}\|\mathbf{x}_{i,j}\|_0 + \sum_{i,j}\|\mathbf{A}\mathbf{x}_{i,j} - \mathbf{R}_{i,j}\mathbf{S}\|^2 \qquad (9)$$

Algorithm 2:

$$\min_{\mathbf{A}\in\mathcal{A},\mathbf{X}} \sum_{i,j} \mu_{ij}\|\mathbf{x}_{i,j}\|_1 + \sum_{i,j}\|\mathbf{A}\mathbf{x}_{i,j} - \mathbf{R}_{i,j}\mathbf{S}\|^2 \qquad (10)$$

3: $\mathbf{S}$ update:

$$\mathbf{S} = (\beta\mathbf{I} + \sum_{i,j}\mathbf{R}_{i,j}^T\mathbf{R}_{i,j})^{-1}(\beta\mathbf{Y} + \sum_{i,j}\mathbf{R}_{i,j}^T\mathbf{A}\mathbf{x}_{i,j}) \qquad (11)$$

The final peak signal-to-noise ratio (PSNR) comparison is summarized in Table 1; and sample images are presented in Figure 1. As can be seen in Table 1, the resulting PSNR values of the proposed algorithm are comparable with the ones obtained by K-SVD. However, visually, K-SVD produces more noticeable artifacts (see the circled spot in Figure 1) than our proposed algorithm. The artifacts may be due to the use of OMP in K-SVD which is less robust to noise than the ℓ_1-regularizer used in Algorithm 2. As for the CPU time, the two algorithms perform similarly in the numerical experiments.

6. REFERENCES

[1] V. A. KotelŠnikov, "On the carrying capacity of the ether and wire in telecommunications," in *Material for the First All-Union Conference on Questions of Communication, Izd. Red. Upr. Svyazi RKKA, Moscow*, 1933.

[2] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[3] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[4] E. T. Whittaker, *On the functions which are represented by the expansions of the interpolation-theory*, Edinburgh University, 1915.

[5] R. Prony, "Essai experimental et analytique sur les lois de la dilatabilite des fluides elastiques et sur celles de la force expansive de la vapeur de leau et de la vapeur de lalkool, r differentes temperatures," *Journal Polytechnique ou Bulletin du Travail fait r Lecole Centrale des Travaux Publics, Paris, Premier Cahier*, pp. 24–76, 1995.

[6] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[8] M. Lustig, D. L. Donoho, and J. M. Pauly, "Rapid mr imaging with compressed sensing and randomly under-sampled 3dft trajectories," in *Proceedings of 14th Annual Meeting of ISMRM*. Citeseer, 2006.

[9] M. Lustig, J. H. Lee, D. L. Donoho, and J. M. Pauly, "Faster imaging with randomly perturbed, under-sampled spirals and l1 reconstruction," in *Proceedings of the 13th Annual Meeting of ISMRM, Miami Beach*, 2005, p. 685.

[10] K. Gedalyahu and Y. C. Eldar, "Time-delay estimation from low-rate samples: A union of subspaces approach," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3017–3031, 2010.

[11] M. Mishali and Y. C. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 993–1009, 2009.

[12] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[13] R. F. Marcia, Z. T. Harmany, and R. M. Willett, "Compressive coded aperture imaging," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 72460G–72460G.

[14] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Asilomar Conference on Signals, Systems, and Computing*, 2005, pp. 1537–1541.

[15] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2580–2594, 2010.

[16] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[17] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[18] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[19] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[20] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," *arXiv preprint arXiv:0809.3083*, 2008.

[21] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[22] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: Design of dictionaries for sparse representation," *Proceedings of SPARS*, vol. 5, pp. 9–12, 2005.

[23] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1999, vol. 5, pp. 2443–2446.

[24] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[25] Z. Jiang, G. Zhang, and L. S. Davis, "Submodular dictionary learning for sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3418–3425.

[26] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[27] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[28] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[29] V. K. Potluru, S. M. Plis, J. L. Roux, B. A. Pearlmutter, V. D. Calhoun, and T. P. Hayes, "Block coordinate descent for sparse nmf," *arXiv preprint arXiv:1301.3527*, 2013.

[30] J. Kim and H. Park, "Sparse nonnegative matrix factorization for clustering," 2008.

[31] D. P. Bertsekas, "Nonlinear programming," 1999.

[32] L. J. Hong, Y. Yang, and L. Zhang, "Sequential convex approximations to joint chance constrained programs: A monte carlo approach," *Operations Research*, vol. 59, no. 3, pp. 617–630, 2011.