Research Article

An Efficient Algorithm for Overcomplete Sparsifying Transform Learning with Signal Denoising

Beiping Hou,¹ Zhihui Zhu,² Gang Li,¹ and Aihua Yu¹

¹The School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou, Zhejiang 310023, China

²The Department of Electrical Engineering and Computer Science, Colorado School of Mines, 1500 Illinois Street, Golden, CO 80401, USA

Correspondence should be addressed to Gang Li; ieligang@zjut.edu.cn

Received 29 March 2016; Accepted 25 May 2016

Academic Editor: Srdjan Stankovic

Copyright © 2016 Beiping Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with the problem of overcomplete transform learning. An alternating minimization based procedure is proposed for solving the formulated sparsifying transform learning problem. A closed-form solution is derived for the minimization involved in transform update stage. Compared with existing ones, our proposed algorithm significantly reduces the computation complexity. Experiments and simulations are carried out with synthetic data and real images to demonstrate the superiority of the proposed approach in terms of the averaged representation and denoising errors, the percentage of successful and meaningful recovery of the analysis dictionary, and, more significantly, the computation efficiency.

1. Introduction

Signal representations play a role of cornerstones in signal processing. They have experienced the evolution from the Fourier transforms, which are traced back to the later part of the 19th century, to the wavelet expansions in the 1980s and then the sparse representations which have been one of the hotspots in signal processing community for the last two decades. Instead of the traditional expansions in terms of bases and frames, sparse redundant representations look for the best approximation of a signal vector with a linear combination of few atoms from an overcomplete set of well-designed vectors [1]. This topic is closely related to the *sparsifying dictionary learning* [2, 3] and the compressed sensing (CS) [4–6], an emerging area of research in which the sparsity of the signals to be recovered is a prerequisite.

Let $v \in \mathbb{R}^{N \times 1}$. The l_p -norm of v is defined as

$$\|v\|_p \triangleq \left(\sum_{k=1}^N |v(k)|^p \right)^{1/p}, \quad p \geq 1. \quad (1)$$

For convenience, $\|v\|_0$ is used to denote the number of nonzero elements in v though l_0 is not a norm as it does not satisfy the homogeneity.

1.1. Signal Models and Dictionary Learning. Let $y = x + \xi \in \mathbb{R}^{N \times 1}$ be the signal under consideration, where x is the clean signal and ξ is an additive noise or modelling error. There are basically two models used for the clean signal x . The first one is called *synthesis model* which assumes that the clean signal x is given by

$$x = \sum_{k=1}^J s(k) \psi_k \triangleq \Psi s, \quad (2)$$

where $\Psi \in \mathbb{R}^{N \times J}$ is usually called a *synthesis dictionary* (throughout this paper, a lower-case letter, say q , is usually used to denote a column vector of the matrix Q . Furthermore, q_k and q_{ij} denote the k th column vector and (i, j) th entry of matrix Q , resp.) and is said *overcomplete* if $N < J$, and $s \in \mathbb{R}^{J \times 1}$ is the coefficient vector of x in Ψ . The signal s is said κ -sparse in Ψ if $\|s\|_0 \leq \kappa$.

The procedure for finding s with x and Ψ given is referred to as *analysis*. Clearly,

$$s = \Omega x \quad (3)$$

is a solution to this problem as long as $\Psi \Omega = I_N$, where I_N is the identity matrix of dimension N . It should be pointed

out that (2) and (3) are equivalent when Ψ is square and nonsingular. When Ψ is overcomplete, such a claim does not hold as (2) is not an one-to-one mapping between x and s . Equation (3) is referred to as *analysis model* [7–12] and $\Omega \in \mathfrak{R}^{J \times N}$ is called *analysis dictionary* or *analysis operator*. When $J > N$, which is assumed throughout this paper, the dictionary is said *redundant* or *overcomplete*.

Let $\{y_n\}$ be a set of signal samples belonging to a certain class of signals. For a given sparsity level, say κ , sparsifying synthesis dictionary learning is to find Ψ such that $y_n = \Psi s_n + \xi_n$, $\forall n$ with $\|s_n\|_0 \leq \kappa$. Methods for learning the dictionary Ψ from signal samples $\{y_k\}$ have been one of the hottest topics in signal processing, among which the method of optimal directions (MOD) [13] and K-singular value decomposition (KSVD) based algorithms [14, 15] have been considered as state of the art in this area. The basic idea behind these methods is based on the alternating minimization strategy [16–18], in which the dictionary Ψ and the sparse coefficients s are updated alternatively using the orthogonal matching pursuit (OMP) techniques [19–21].

Compared with the synthesis model-based sparsifying dictionary learning, its analysis model-based counterpart has received less attention. The analysis dictionary learning problem can be stated as

$$\begin{aligned} \{\Omega, X\} \triangleq \arg \min_{\Omega, \tilde{X}} \quad & \|Y - \tilde{X}\|_F^2, \\ & \|\tilde{\Omega} \tilde{x}_m\|_0 \leq J - l \triangleq \kappa, \quad \forall m, \end{aligned} \quad (4)$$

where l , denoting the number of zero elements in s , is the cosparsity given and $\tilde{x}_m$ denotes the m th column of $\tilde{X}$. (throughout this paper, we use “tilde” (say $\tilde{Q}$) to denote a (matrix/vector) variable, while Q is used for the optimal/true one in a problem). As X is the clean signal, (4) is also signal denoising.

The first attempt to address such a problem was given in [7], where it is suggested to incrementally learn the dictionary Ω one row at a time. The performance of the proposed algorithm, however, depends heavily on a randomized initialization strategy. A different approach to the dictionary learning was proposed in [8–10]. As proposed in [10], (4) is investigated with the following formulation:

$$\begin{aligned} \min_{\tilde{\Omega}, \tilde{X}} \quad & \|\tilde{\Omega} \tilde{X}\|_1 \\ \text{s.t.} \quad & \tilde{\Omega} \in \mathcal{E}, \\ & \|Y - \tilde{X}\|_F \leq \epsilon_0, \end{aligned} \quad (5)$$

where $\mathcal{E}$ is a specified set within which the optimal operator is searched and $\epsilon_0 \geq 0$ is the parameter corresponding to the noise level. Such a problem is alternatively addressed using a regularized version in [10].

Another signal model, which is very much close to the analysis model, is the *transform model* raised in [22]. Such a model is characterized with

$$Wy = s + \epsilon, \quad (6)$$

where $W \in \mathfrak{R}^{J \times N}$ is called a *transform* and s is sparse with $\|s\|_0 \leq \kappa$, while ϵ is the representation/modeling error in transform domain and is assumed to be *very small*. It is interesting to note that, for an analysis signal $y = x + \xi$ with $\Omega x = s$ sparse, one has $\Omega y = s + \Omega \xi$ which is always in the form specified in (6), while if y belongs to the transform model-based class in Ω , which means there exists a pair (s, e) with s being sparse such that $\Omega y = s + e$, $\Omega [x \ \xi] = [s \ e]$ may have no solution for x, ξ at all given that the number of equations is bigger than that of unknowns due to $N < J$ assumed.

When W is available, the signal y is transformed to κ -sparse vector s :

$$\begin{aligned} s \triangleq \arg \min_{\tilde{s}} \quad & \|\tilde{s} - Wy\|_2^2, \\ \text{s.t.} \quad & \|\tilde{s}\|_0 \leq \kappa. \end{aligned} \quad (7)$$

The solution can be achieved simply by thresholding the product Wy and retaining only the κ largest coefficients (magnitude-wise) of this product. The estimate $\hat{y}$ of the original signal y can be found from s and W by minimizing $\|s - W\tilde{y}\|_2^2$ with respect to $\tilde{y}$. This yields

$$\hat{y} = (W^T W)^{-1} W^T s \quad (8)$$

as long as W is full rank (under the assumption that $J \geq N$), where T denotes the transpose operator.

The sparsifying transform learning problem is formulated as

$$\begin{aligned} \{W, S\} \triangleq \arg \min_{W, \tilde{S}} \quad & \|\tilde{S} - \tilde{W}Y\|_F^2 \\ & \|\tilde{s}_m\|_0 \leq \kappa, \quad \forall m. \end{aligned} \quad (9)$$

This problem is not well defined and has the problem of so-called *scale ambiguity*. It is observed that a trivial solution for the above problem is $S = \mathbf{0}$ and $W = \mathbf{0}$. One approach to prevent such solutions is to add some penalty functions into the objective. In [22], the penalty $\nu \|\tilde{W}\|_F^2 - \log \det(\tilde{W})$ was included in the cost function for learning a complete transform with dimension $N \times N$. The term $\log \det(\tilde{W})$ is used to force the full rank of W and the term $\|\tilde{W}\|_F^2$ is applied to avoid the scaling issue. See

$$\begin{aligned} \mathcal{L}_{RB}(\tilde{W}, \tilde{S}) \triangleq \quad & \|\tilde{S} - \tilde{W}Y\|_F^2 \\ & + \lambda \left[\nu \|\tilde{W}\|_F^2 - \log \det(\tilde{W}) \right], \end{aligned} \quad (10)$$

where λ and ν are two positive constants, while $\log \det(\cdot)$ is the natural logarithmic operator applied to the determinant of a matrix. The optimal transform design was then reformulated as

$$\begin{aligned} \{W, S\} \triangleq \arg \min_{W, \tilde{S}} \quad & \mathcal{L}_{RB}(\tilde{W}, \tilde{S}), \\ & \|\tilde{s}_m\|_0 \leq \kappa, \quad \forall m \end{aligned} \quad (11)$$

subject to $\det(W) > 0$. Such a problem was solved in [22, 23].

1.2. Contributions. In this paper, we will consider solving (9) with given (Y, κ) . An efficient algorithm is proposed to solve the optimal sparsifying transform design problem. It should be pointed out that our proposed approach is more general than those in [22, 23] as it deals with overcomplete transform learning and is more efficient than [24] since the transform is updated using a closed-form solution rather than the one obtained with a gradient-based numerical procedure. All of these will be elaborated further in next section. Roughly speaking, (4) has been attacked using alternative optimization, in which with $\tilde{X}$ obtained, Ω is updated based on $\|\tilde{\Omega}\tilde{x}_m\|_0 \leq J - l \triangleq \kappa, \forall m$. This can be considered with

$$\begin{aligned} \Omega &\triangleq \arg \min_{\tilde{\Omega}, \tilde{S}} \|\tilde{S} - \tilde{\Omega}\tilde{X}\|_F^2, \\ \|\tilde{S}_m\|_0 &\leq \kappa, \quad \forall m. \end{aligned} \quad (12)$$

As seen, this problem is exactly the same as (9) with Y replaced with $\tilde{X}$. So, the efficiency of the analysis KSVD proposed in [12] is expected to be improved using our proposed algorithm for analysis dictionary update. We will demonstrate the potential of this proposed algorithm in a series of experiments using both synthetic and real data and compare it with the approach in [12]. As to be seen, our approach yields a performance comparable to that by the latter in terms of denoising and recovery of the reference dictionary and more importantly a much improved computation efficiency.

The outline of this paper is given as follows. In Section 2, after briefing some related work, we formulate the problem to be investigated in the context of transform learning and how it is related to analysis dictionary learning. Section 3 is devoted to solving the optimal transform design. An efficient algorithm is derived for that purpose, in which a closed-form solution is obtained for updating the transform. The performance of this algorithm and its application to signal denoising are discussed in this section. In order to demonstrate the superiority of the proposed algorithm, experiments are carried out in Section 4, which show promising performance of the proposed approach. Some concluding remarks are given in Section 5.

2. Preliminaries and Problem Formulation

2.1. Related Works. The first attempt to address the analysis dictionary learning was given in [7], where it is suggested to incrementally learn the dictionary Ω one row at a time. The performance of the proposed algorithm, however, depends heavily on a randomized initialization strategy. A different approach to the dictionary learning was proposed in [8–10], where the goal of sparsifying the representations $\Omega\tilde{X}$ is formulated using an l_1 -based penalty function rather than the l_0 -based one. To avoid the trivial solutions, the dictionary is constrained to be uniform normalized tight frame; that is, $\Omega^\top \Omega = I_N$, the identity matrix of dimension N . Such a constraint clearly limits the searching space for the optimal dictionary.

Recently, it was argued in [12] that, in the analysis model, one should emphasize the zeros of Ωx . Based on this argument, (4) was reformulated in [12] as

$$\begin{aligned} \{\Omega, X, \{\Lambda_i\}_{i=1}^L\} &\triangleq \arg \min_{\tilde{\Omega}, \tilde{X}, \{\tilde{\Lambda}_i\}_{i=1}^L} \|Y - \tilde{X}\|_F^2, \\ \tilde{\Omega}_{\tilde{\Lambda}_i} \tilde{x}_i &= \mathbf{0}, \\ \text{Rank}(\tilde{\Omega}_{\tilde{\Lambda}_i}) &= N - r, \\ i &= 1, \dots, L, \\ \|\tilde{\omega}_m^\dagger\|_2 &= 1, \quad \forall m, \end{aligned} \quad (13)$$

where $q_m^\dagger$ denotes the m th row vector of matrix Q , $\tilde{\Lambda}_i$ is the *cosupport* of $\tilde{x}_i$, defined as the set of indices for the l rows of $\tilde{\Omega}$ that are orthogonal to $\tilde{x}_i$, and $\tilde{\Omega}_{\tilde{\Lambda}_i}$ is the submatrix of $\tilde{\Omega}$ that contains only the rows indexed by $\tilde{\Lambda}_i$, while r is the dimension of the subspace to which $\tilde{x}_i$ belongs.

An algorithm, called analysis KSVD, was proposed in [12] to address (13). The analysis KSVD, denoted as AKSVD, was developed in a similar way as the one adopted by its counterpart KSVD [14] for the synthesis model and as shown in [12], outperforming the one reported in [7]. It has been observed, however, that the AKSVD is very time consuming and therefore more efficient algorithms are needed.

In order to restrict the solution set of the transform learning problem (9) to an admissible set, $\tilde{W}^\top \tilde{W} = I_J$ was applied in [25], which means forcing the row of the transform $\tilde{W}$ to be a tight frame. However, only tight frame constraint is insufficient to prevent undesired solutions of (9) for learning an overcomplete transform $\tilde{W}$ ($J > N$), for which it is highly possible that practical algorithms result in a local minimal solution of the form $\tilde{W} = [\tilde{W}_1^\top \mathbf{0}^\top]^\top$ with $\tilde{W}_1$ being a tight frame. In [24], the overcomplete transform learning (9) was attacked with the following penalty function to be embedded to the cost function:

$$\nu \left(\|\tilde{W}\|_F^2 - \log \left(\tilde{W}^\top \tilde{W} \right) \right) + \eta \sum_{n \neq n'} |\tilde{w}_n^\dagger \tilde{w}_{n'}^\dagger|^q, \quad (14)$$

where ν and η are weighting factors and q is usually set to 1 and 2. An iterative procedure using the alternating minimization strategy was proposed to solve the corresponding problem, where a gradient-based algorithm is utilized for updating the transform. Clearly, such a method is time consuming and may suffer from local minimum issue due to the use of gradient-based algorithm. Furthermore, the possibility of having same rows in $\tilde{W}$ can be reduced with the second term of the penalty but zero rows may occur if $\tilde{W}$ is not (row) normalized.

2.2. Problem Formulation. Based on the previous discussions, the optimal sparsifying transform learning problem can be formulated as

$$\begin{aligned} \{W, S\} &\triangleq \arg \min_{\widetilde{W}, \widetilde{S}} \|\widetilde{S} - \widetilde{W}Y\|_F^2, \\ \|\widetilde{s}_m\|_0 &\leq \kappa, \quad \forall m, \\ \widetilde{W}^\mathcal{T} \widetilde{W} &> 0 \end{aligned} \quad (15)$$

subject to

$$\|\widetilde{w}_k^\dagger\|_0 = 1, \quad \forall k. \quad (16)$$

To implement (15), we consider the following cost function:

$$\mathcal{L}(\widetilde{W}, \widetilde{S}) \triangleq \|\widetilde{S} - \widetilde{W}Y\|_F^2 + \lambda \mathcal{F}(\widetilde{W}), \quad (17)$$

where $\mathcal{F}(\widetilde{W})$ is a penalty term (i.e., a regularizer) defined as

$$\mathcal{F}(\widetilde{W}) \triangleq \nu \|\widetilde{W}\|_F^2 - \log \det(\widetilde{W}^\mathcal{T} \widetilde{W}). \quad (18)$$

Both λ and ν are positive parameters to be discussed later.

The problem to be addressed in this paper is finally formulated as

$$\begin{aligned} \{W, S\} &= \arg \min_{\widetilde{W}, \widetilde{S}} \mathcal{L}(\widetilde{W}, \widetilde{S}), \\ \|\widetilde{s}_m\|_0 &\leq \kappa, \quad \forall m, \\ \|\widetilde{w}_k^\dagger\|_2 &= 1, \quad \forall k. \end{aligned} \quad (19)$$

Comment 1. (i) The additional regularizer $\mathcal{F}(\widetilde{W})$ in (17) serves as a constraint on the space for searching the true transform W . The term $\|\widetilde{W}\|_F^2$ is to ensure that W is not too big, while $-\log \det(\widetilde{W}^\mathcal{T} \widetilde{W})$ is to guarantee that W is not too small and hence the constraint of full rank is met. The combination of the two terms ensures that the optimal dictionary is searched within a well behaved space such that the undesired scaling problem can be avoided.

(ii) As to be seen in the next subsection, the minimum of this regularizer is a set of tight frames. Therefore, if (4) has a solution W close to a tight frame, it can be well estimated by (19).

(iii) Comparing (19) with (11), one realizes that (19) can be used for optimal overcomplete sparsifying transform learning. Besides, instead of $-\log \det(\widetilde{W})$, $-\log \det(\widetilde{W}^\mathcal{T} \widetilde{W})$ is adopted in our approach. By doing so, the constraint $\det(\widetilde{W}) > 0$ is no longer needed.

2.3. Analysis Dictionaries and Transforms. Consider the analysis model-based dictionary learning (4). Define $E \triangleq Y - \widetilde{X}$ and $\mathcal{E} \triangleq \widetilde{\Omega}Y - \widetilde{S}$ as the *representation error* and the *sparsification error*, respectively, where $\widetilde{\Omega}\widetilde{X} = \widetilde{S}$. Then

$$\widetilde{\Omega}Y - \widetilde{S} = \widetilde{\Omega}(Y - \widetilde{X}). \quad (20)$$

It is assumed in the sequel that the true dictionary Ω is full rank. Such an assumption ensures that the row vectors of Ω can span the entire signal space $\mathfrak{R}^{N \times 1}$.

The following lemma specifies a class of true dictionaries Ω for which the equivalence between (15) and (4) holds.

Lemma 1. Equation (15) is equivalent to (4) if the true dictionary Ω is a c -tight frame (TF) ($c > 0$); that is, Ω has an SVD of form

$$\Omega = cU [I_N \quad \mathbf{0}]^\mathcal{T} V^\mathcal{T}, \quad (21)$$

where both U and V are orthonormal matrices of proper dimension.

Proof. With (20) and (21), we have

$$\begin{aligned} \|\widetilde{S} - \Omega Y\|_F^2 &= \|\Omega(Y - \widetilde{X})\|_F^2 \\ &= c^2 \|U [I_N \quad \mathbf{0}]^\mathcal{T} V^\mathcal{T} (Y - \widetilde{X})\|_F^2 \\ &= c^2 \|[I_N \quad \mathbf{0}]^\mathcal{T} V^\mathcal{T} (Y - \widetilde{X})\|_F^2 \\ &= c^2 \|V^\mathcal{T} (Y - \widetilde{X})\|_F^2 = c^2 \|Y - \widetilde{X}\|_F^2. \end{aligned} \quad (22)$$

Then, Lemma 1 follows immediately from the fact that the two constraints $\|\widetilde{s}_m\|_0 \leq \kappa$ and $\|\Omega \widetilde{x}_m\|_0 \leq \kappa$ are equivalent as $\widetilde{S} = \Omega \widetilde{X}$. $\square$

Comment 2. As seen from Lemma 1, the equivalence between the two problems holds if the true dictionary Ω is a c -tight frame. This is in general not true. However, it is observed that for given Y the problem defined by (4) has in general more than one solution (Ω, X) . In fact, define $D_\alpha \triangleq \text{diag}(\alpha_1, \dots, \alpha_m, \dots, \alpha_l)$ and $D_\beta \triangleq \text{diag}(\beta_1, \dots, \beta_m, \dots, \beta_l)$. It can be seen that if (Ω_0, X_0) is a solution, is (Ω, X) :

$$\begin{aligned} \Omega &= D_\alpha \Omega_0 T, \\ X &= T^{-1} X_0 D_\beta, \end{aligned} \quad (23)$$

where D_α is any given nonsingular diagonal matrix, while both $T \in \mathfrak{R}^{N \times N}$ and D_β are arbitrary except $\det(D_\beta) \neq 0$ such that $\|Y - X\|_F = \|Y - X_0\|_F$. This means that there exist a lot of degrees of freedom in the solution set of (4) and that by forcing finding the solution for which the dictionary Ω is the closest to a TF, (15) is then expected to yield a transform W that is a good estimate of a solution for the analysis dictionary learning problem (4).

2.4. Rule of Thumb for Choosing λ and ν . Now, we present a result regarding minimization of $\mathcal{F}(\widetilde{W})$. A similar result was given in [22].

Lemma 2. Let $\mathcal{F}(\widetilde{W})$ be defined in (18). Then,

$$\mathcal{F}(\widetilde{W}) \geq N [1 + \log(\nu)] \triangleq \eta_0 \quad (24)$$

with the equality achieved if and only if $\widetilde{W}$ is a $\sqrt{\nu^{-1}}$ -tight frame.

Proof. Let $\tilde{W} = U [\Sigma_N \mathbf{0}]^{\mathcal{T}} V^{\mathcal{T}}$ be the SVD of $\tilde{\Omega}$, where $\Sigma_N = \text{diag}(\sigma_1, \dots, \sigma_k, \dots, \sigma_N)$ with $\sigma_k \geq \sigma_{k+1} > 0, \forall k$. Clearly, $\mathcal{F}(\tilde{\Omega}) = \sum_{k=1}^N [\nu \sigma_k^2 - \log(\sigma_k^2)] \triangleq \sum_{k=1}^N f(\sigma_k)$.

Note that $df(\sigma_k)/d\sigma_k = \nu 2\sigma_k - 2\sigma_k/\sigma_k^2$ and $\sigma_k > 0$. It follows from $df(\sigma_k)/d\sigma_k = 0$ that the solution for minimizing $f(\sigma_k)$ is given by

$$\sigma_k^* = \sqrt{\nu^{-1}} \implies f(\sigma_k^*) = 1 + \log(\nu) \leq f(\sigma_k), \quad \forall k \quad (25)$$

and hence (24) holds. Clearly, the equality is attained *if and only if* $\sigma_k^* = \sqrt{\nu^{-1}}, \forall k$. $\square$

Comment 3. (i) Based on Lemma 2, the cost function $\mathcal{L}(\tilde{W}, \tilde{S}) = \|\tilde{S} - \tilde{W}Y\|_F^2 + \lambda \mathcal{F}(\tilde{W})$ is bounded from below with $\lambda \eta_0$.

(ii) It follows from Lemmas 1 and 2 that if the reference analysis operator happens to be a tight frame, then the solution of (19) should yield the same operator no matter what λ is taken.

(iii) As indicated in Comment 2, the solution W can be expected to be close to a tight frame. This suggests that λ should take a value much larger than one since when λ is very large, minimizing $\mathcal{L}(\tilde{W}, \tilde{S})$ can be approximated with minimizing $\mathcal{F}(\tilde{W})$.

(iv) As to be seen from (19), we need to normalize the dictionary $\tilde{W}$ such that $\|\tilde{w}_j^\dagger\|_2 = 1, \forall j$. Under this constraint, the constant c in (21) turns out to be $\sqrt{J/N}$. Since $J \geq N$ holds for overcomplete analysis dictionary, $\nu < 1$ should be ensured.

Before turning to next section, we should point out that though the proposed approach is not directly related to the overcomplete analysis dictionary learning (4) as (13) is in general, it can provide a good estimate of the true dictionary in a very efficient way, which is believed to be useful in improving the analysis KSVD algorithm.

3. The Proposed Algorithm

This section is devoted to attacking the optimal sparsifying transform learning problem by deriving an algorithm to solve (19).

3.1. The Algorithm. Note that the problem defined in (19) is nonconvex. A practical approach to such problems is to adopt the alternating minimization strategy [16–18]. The basic idea is to update the dictionary $\tilde{W}$ and the (column) sparse matrix $\tilde{S}$ alternatively. The proposed algorithm is outlined below.

Alg_{Proposed}

Initialization. Set the training data matrix Y , the number of iterations N_{ite} , and the parameters λ, ν ; initiate W with W_0 . (there are several ways to initialize W . A simple way is to set $W = \text{randn}(J, N)$. Other way is that for the m th row ($m = 1, 2, \dots, J$) we first randomly choose $N - 1$ columns from the training data matrix Y to form a submatrix $\bar{Y}$, then set $w_m^\dagger = e - e\bar{Y}\bar{Y}^\dagger$, where $e = \text{randn}(1, N)$ and $\bar{Y}^\dagger$ denotes

the pseudoinverse of $\bar{Y}$. At last we normalize each row to unit l_2 norm. In the simulations to be presented, we choose the second approach to initialize W in order to have a fair comparison with [12]).

Begin ($k = 1, 2, \dots, N_{\text{ite}}$)

Step 1. Update S such that $\mathcal{L}(W_{k-1}, \tilde{S})$ is minimized under the constraint $\|\tilde{s}_m\|_0 \leq \kappa, \forall m$. This is equivalent to

$$S_k \triangleq \arg \min_{\tilde{S}} \|\tilde{S} - W_{k-1}Y\|_F^2, \quad (26)$$

$$\|\tilde{s}_m\|_0 \leq \kappa, \quad \forall m.$$

Step 2. With S_k just obtained, update W by solving

$$\tilde{W}_k \triangleq \arg \min_{\tilde{W}} \mathcal{L}(\tilde{W}, S_k). \quad (27)$$

Step 3. Normalize $\tilde{W}_k$ such that all the rows of $W_k = T_{sc} \tilde{W}_k$ are of unit length. This can be achieved by setting the diagonal T_{sc} with its (j, j) th entry equal to the inverse of the length for the j th row of $\tilde{W}_k$ for $j = 1, 2, \dots, J$.

End. Output $\hat{W} = W_{N_{\text{ite}}}$, the estimate of the true dictionary W .

Now, let us elaborate the three steps involved in the proposed algorithm further.

The first one involves updating S with (26), whose solution S_k , as mentioned before, can be achieved directly with the product $V \triangleq W_{k-1}Y$ in the following way.

Begin ($m = 1, 2, \dots, L$)

(i) Determine ϱ_m which is equal to the absolute value of the κ th largest coefficients (magnitudewise) in the m th column of V .

(ii) For $j = 1, 2, \dots, J$, set the (j, m) th element of S_k to

$$\begin{cases} v_{jm}, & |v_{jm}| \geq \varrho_m, \\ 0, & |v_{jm}| < \varrho_m, \end{cases} \quad (28)$$

where v_{jm} denotes the (j, m) th element of matrix V .

End. Output S_k . (In case where there are more than κ nonzero elements in S_k , just keep any set of the first κ largest ones and set the others to zero.)

Now, let us consider (27) involved in Step 2. Define

$$M \triangleq \tilde{S}Y^{\mathcal{T}}, \quad (29)$$

$$Z \triangleq \lambda \nu I_N + YY^{\mathcal{T}}.$$

It can be shown that the cost function $\mathcal{L}(\tilde{W}, \tilde{S})$ can be rewritten as

$$\begin{aligned} \mathcal{L}(\tilde{W}, \tilde{S}) &= \|\tilde{S}\|_F^2 - 2 \text{tr} [M\tilde{W}^{\mathcal{T}}] + \text{tr} [\tilde{W}Z\tilde{W}^{\mathcal{T}}] \\ &\quad - \lambda \log \det (\tilde{W}^{\mathcal{T}}\tilde{W}) \triangleq \mathcal{L}_1(\tilde{W}), \end{aligned} \quad (30)$$

where $\text{tr}[\cdot]$ denotes the trace operator of a matrix and $\mathcal{L}_1(\tilde{W})$ is used to emphasize the fact that $\tilde{S}$ is fixed in the minimization to be considered.

Therefore, (27) is just a special case of the following minimization problem:

$$W^* \triangleq \arg \min_{\tilde{W}} \mathcal{L}_1(\tilde{W}). \quad (31)$$

Theorem 3. Let $M \in \mathbb{R}^{J \times N}$, $Z \in \mathbb{R}^{N \times N}$, and $\tilde{S}$ three matrices independent of matrix $\tilde{W} \in \mathbb{R}^{J \times N}$ with $J \geq N$, and $Z = U_z \Lambda^2 U_z^\mathcal{T}$ an SVD of the (symmetric) positive-definite Z and $Z_{\text{sqr}} \triangleq U_z \Lambda U_z^\mathcal{T}$. Furthermore, let $\tilde{M} \triangleq M Z_{\text{sqr}}^{-1} = U_0 \begin{bmatrix} \Pi^2 \\ \mathbf{0} \end{bmatrix} V_0^\mathcal{T}$ be an SVD of $\tilde{M}$, where $\Pi = \text{diag}(\pi_1, \dots, \pi_k, \dots, \pi_N)$. A solution to (31) is then given by

$$W^* = U_0 \begin{bmatrix} \Sigma^2 \\ \mathbf{0} \end{bmatrix} V_0^\mathcal{T} Z_{\text{sqr}}^{-1}, \quad (32)$$

where $\Sigma^2 = \text{diag}(\sigma_1^2, \dots, \sigma_k^2, \dots, \sigma_N^2)$ with

$$\sigma_k^2 = \frac{\pi_k^2 + \sqrt{\pi_k^4 + 4\lambda}}{2}, \quad \forall k. \quad (33)$$

The detailed proof can be found in Appendix.

As seen from (29), no matter what Y is taken the corresponding Z matrix in our problem is always (symmetric) positive-definite as long as $\lambda > 0$ and $\nu > 0$. Therefore, (27) in Step 2 of the proposed algorithm can be solved using Theorem 3.

Step 3 is to normalize the transform such that W_k has all its rows of unit length. This is for avoiding zero rows in the transform W and the *scaling ambiguity* [22]. Such scaling has been popularly used in sparsifying dictionary design.

Comment 4. As noted, the proposed algorithm shares the same framework as the one used in [22]. It should be pointed out that besides the fact that our algorithm is more general in the sense that it can deal with analysis dictionaries (transforms) of any dimension rather those square nonsingular ones, the main difference between the two lies in the way of how to solve (27) in Step 2. In [22], a gradient-based algorithm was used to attack (27). Such an algorithm suffers in general from slow convergence and more seriously convergence to a local minimum of the problem. Therefore, our proposed algorithm is more efficient and faster convergence is then expected.

3.2. Convergence. Consider the situation when the row normalization (i.e., Step 3) is not executed in proposed $\text{Alg}_{\text{Proposed}}$. As the solutions for the minimization involved in both Steps 1 and 2 are exact, the cost function $\mathcal{L}(\tilde{W}, \tilde{S})$ is ensured to be decreasing. Given that the cost function is bounded from below, the proposed $\text{Alg}_{\text{Proposed}}$ without Step 3 can ensure that the cost function $\mathcal{L}(\tilde{W}, \tilde{S})$ converges. As mentioned before, the obtained transform W may have zero rows during the iterative procedure. In order to avoid this

phenomenon, Step 3 has to be applied when the transform is overcomplete and, in that case, the cost function is found empirically to converge as well.

As argued in [22], the convergence of cost function does not indicate that of the iterates. There have been no theoretical results achieved regarding convergence of the iterates for the proposed algorithm. However, extensive experiments showed that the proposed algorithm does demonstrate good behavior of convergence.

3.3. Computation Efficiency. A thorough comparison between the algorithm proposed in [22], denoted as Alg_{RB} , and some relevant algorithms in terms of computational cost can be found in [22]. Since our proposed algorithm shares the same framework with Alg_{RB} and is derived as an alternative of $\text{Alg}_{\text{AKSVD}}$, the one recently reported in [12], we mainly focus on comparing our proposed $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$. For the sake of completeness, we present the computation issues of our proposed $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$ as follows.

Our algorithm is an iterative method. In each iteration, there are three steps. They are sparse coding, dictionary updating, and the dictionary normalizing. In the sparse coding step, we simply keep the κ largest entries of each column in the matrix WY by full sorting. This, as indicated in [26], requires $O(JL \log N)$ operations. Knowing that computing ΩY requires $O(JNL)$ operations, one concludes that the sparse coding step needs roughly $O(JL(N + \log N))$ flops. In the dictionary update step, what we mainly do is just computing the SVD form of a matrix M . Thus, this step costs approximately $O(NJ^2 + N^3)$ flops, while in the normalization step, it needs $O(JN)$ flops. Thus the total cost per iteration of the proposed algorithm is the sum of the three, which is dominated by $O(JNL)$ under the assumption of $N \leq J \ll L$.

As to $\text{Alg}_{\text{AKSVD}}$, there are two steps involved in each iteration, sparse coding and dictionary updating, and the computation cost is dominated by the sparse coding step. As given in [12], the sparse coding step requires $O(LN^3J)$ operations if the optimized backward greedy (OBG) algorithm is utilized. Therefore, the overall computation complexity scales as $O(LN^3J)$.

Based on the computations above, one can see that our proposed algorithm is much more efficient than $\text{Alg}_{\text{AKSVD}}$. This is confirmed by experiments to be given in the next section.

3.4. Signal Denoising. For a measurement y_i and a (row normalized) dictionary $\hat{\Omega}$ given, the clean signal x_i is estimated in $\text{Alg}_{\text{AKSVD}}$ via solving the problem

$$\begin{aligned} \{\hat{x}_i, \hat{\Lambda}_i\} &\triangleq \arg \min_{x, \Lambda} \|x - y_i\|_2, \\ \hat{\Omega}_\Lambda x &= 0, \\ \text{Rank}(\hat{\Omega}_\Lambda) &= N - r, \end{aligned} \quad (34)$$

where $\Lambda \subset \{1, 2, \dots, L\}$ is the set of row indices and r is the dimension of the subspace to which the signal belongs. The OBG is used to attack such a problem and it has to be run

L times for each iteration in order to obtain the clear signal matrix X and the final estimate of the true dictionary Ω . That is why Alg_{AKSVD} is very time consuming.

In Alg_{Proposed}, instead of X the (column) sparse matrix S is used. As mentioned before, the latter can be updated efficiently using (28). Taking the transform obtained using Alg_{Proposed} as the analysis dictionary, we can run the OBG to get the corresponding estimate $\hat{x}_i$ of x_i for $i = 1, 2, \dots, L$.

Alternatively, following the same approach as in [22] we can consider the following:

$$\begin{aligned} \{W, X, S\} = \arg \min_{\tilde{W}, \tilde{X}, \tilde{S}} \mathcal{J}(\tilde{W}, \tilde{X}, \tilde{S}), \\ \|\tilde{s}_m\|_0 \leq \kappa, \quad \forall m, \end{aligned} \quad (35)$$

where the objective function $\mathcal{J}(\tilde{W}, \tilde{X}, \tilde{S})$ is defined as

$$\mathcal{J}(\tilde{W}, \tilde{X}, \tilde{S}) \triangleq \|\tilde{W}\tilde{X} - \tilde{S}\|_F^2 + \lambda \mathcal{F}(\tilde{W}) + \mu \|Y - \tilde{X}\|_F^2 \quad (36)$$

with $\mathcal{F}(\tilde{W})$ the same as defined before and $\mu > 0$ constant determining the importance of the representation error.

This is simultaneous optimization of the dictionary W , the clean signals X , and the sparse coefficients S . It is easy to understand that this problem can be addressed using alternating minimization and a similar algorithm to Alg_{Proposed} can be derived. In fact, such an algorithm can be obtained from Alg_{Proposed} with adding (i) $X_0 = Y$ to the *initialization* and (ii) a new step between Steps 1 and 2 for updating the clean signals X :

$$X_k \triangleq \arg \min_{\tilde{X}} \mathcal{J}(W_{k-1}, \tilde{X}, S_k). \quad (37)$$

The above is equivalent to minimizing $\|W_{k-1}\tilde{X} - S_k\|_F^2 + \mu \|Y - \tilde{X}\|_F^2$ with respect to $\tilde{X}$, a standard least squares problem.

In this paper, we focus on the OBG based denoising approach as comparisons are given between proposed Alg_{Proposed} and Alg_{AKSVD}.

4. Experimental Results

In this section, we present some experiments to illustrate the performance of our proposed algorithm Alg_{Proposed} and compare it with that by Alg_{AKSVD} given in [12]. For convenience, the dictionaries, obtained using these two algorithms, are denoted with Ω_{Proposed} and Ω_{AKSVD} , respectively. All the experiments are performed on a computer with an Intel Core i7 CPU at 3.40 GHz, 16 GB memory, and 64-bit Windows 7 operating system.

4.1. Synthetic Experiments. In order to have a fair comparison, the setup used here is similar to [12]. Let $J = 50$, $N = 25$, and $L = 50,000$. A $J \times N$ matrix W is generated with normally distributed entries and then $\Omega = D_\alpha W$, where D_α is such a diagonal matrix that all the rows of Ω have unit l_2 -norm. This Ω serves as the true overcomplete analysis dictionary. Then, a set of $\mathcal{L}$ vectors $\{x_i\}$ of $N \times 1$ is produced with each living in a 4-dimensional subspace. Each

FIGURE 1: Histogram of the effective sparsity of $\{\Omega x_i\}$ ($i = 1, 2, \dots, L$) for $J = 50$, $N = 25$, and $L = 50000$.

vector x_i is obtained as follows: we first construct a submatrix $\Omega^{(i)} \in \mathbb{R}^{21 \times N}$ by randomly choosing 21 rows from Ω and then compute the orthonormal basis $\{\psi_k^{(i)}\}$ for the null space of $\Omega^{(i)}$, which can be obtained from the SVD of $\Omega^{(i)}$. With $\Psi^{(i)} \triangleq [\psi_1^{(i)} \ \psi_2^{(i)} \ \psi_3^{(i)} \ \psi_4^{(i)}]$, calculate $x_i = \Psi^{(i)} s_i$, where each element in s_i is randomly generated by Gaussian distribution of independent and identically distributed (i.i.d.) zero-mean and unit variance. At last, each vector x_i is normalized so that $\|x_i\|_2 = 1$ for all $i = 1, 2, \dots, L$. We can see that Ωx_i has at least 21 zero components for all $\{x_i\}$ ($i = 1, 2, \dots, L$) generated in such a way.

Let $\mathcal{F}_\kappa : \mathbb{R}^J \rightarrow \mathbb{R}^J$ denote a hard thresholding operator that keeps the largest (in magnitude) κ elements of a vector. We define the effective sparsity $\tilde{\kappa}$ of a length- $\mathcal{J}$ vector s as

$$\begin{aligned} \tilde{\kappa} = \min \quad & \kappa \\ \text{s.t.} \quad & \|\mathcal{F}_\kappa(s)\|_2 \geq 0.99 \|s\|_2. \end{aligned} \quad (38)$$

Figure 1 represents the histogram of the effective sparsity level of $\{\Omega x_i\}$ ($i = 1, 2, \dots, L$). As can be seen, the effective sparsity level is varying from 17 to 25 and is most concentrated near 21.

The measurements $\{y_i\}$ are generated with

$$y_i = x_i + \epsilon_i, \quad \forall i, \quad (39)$$

where each entry of the additive noise vector ϵ_i is a white Gaussian noise with i.i.d. zero-mean and σ^2 -variance. Define the signal-to-noise ratio (SNR) as $\rho_{\text{snr}} \triangleq 10 \log_{10}(E[\|x_i\|_2^2]/E[\|\epsilon_i\|_2^2])$, where $E[\cdot]$ denotes the mathematical expectation operator. When $\sigma = 0.2/\sqrt{N} = 0.04$, $\rho_{\text{snr}} = 10 \log_{10}(1/(N * \sigma^2)) = 10 \log_{10}(25) \approx 14$ dB.

We then estimate the analysis dictionary Ω using Alg_{AKSVD} and Alg_{Proposed} with $\tilde{X} = Y$ (see (12)), respectively, for both noise-free case (i.e., $\sigma = 0$) and noisy case with noise level $\rho_{\text{snr}} = 14$ dB. In both cases, we fix $\kappa = 21$ and $N_{\text{ite}} = 200$, while $\nu = 0.01$ and $\lambda = 300$ for Alg_{Proposed}. (As we said before,

FIGURE 2: Evolution of the objective function $\mathcal{L}(\Omega_k, S_k)$ for $J = 50$, $N = 25$, and $L = 50000$.

Ωx_i has at least 21 zero components for all $\{x_i\}$. Thus, we can choose κ such that $\kappa \leq J - 21 = 29$. It is illustrated in Figure 1 that the effective sparsity is concentrated near 21, and therefore we choose $\kappa = 21$. Both $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$ are initialized with a dictionary Ω_0 in which each row is generated as follows: we first randomly choose $N - 1$ columns from the training data matrix Y to form a submatrix $\bar{Y}$; then set $\omega_m^\dagger = e - e\bar{Y}\bar{Y}^\dagger$, where $e = \text{randn}(1, N)$ and $\bar{Y}^\dagger$, as defined before, denotes the pseudoinverse of $\bar{Y}$; at last we normalize each row to unit l_2 norm [12].

Convergence Behavior of $\text{Alg}_{\text{Proposed}}$. The convergence behavior of our proposed $\text{Alg}_{\text{Proposed}}$ is shown in Figures 2 and 3 in terms of the objective function $\mathcal{L}(\Omega_k, S_k)$ and the relative change of iterates which is defined as $\|\Omega_k - \Omega_{k-1}\|_F / \|\Omega_{k-1}\|_F$.

Comment 5. (i) The results are self-explanatory. As expected, the objective function $\mathcal{L}(\Omega_k, S_k)$ decreases smoothly in both noise-free case and noisy case. This is confirmed in Figure 2, from which it is seen that the latter case yields larger $\mathcal{L}(\Omega_k, S_k)$ than the noise-free case because of the larger sparsification error $\|S_k - \Omega_k Y\|_F^2$ in the latter case.

(ii) The convergence behavior of the iterates Ω_k is depicted in Figure 3. One observes that the iterates of the proposed algorithm show a very good trend to convergence in both cases.

Figures 4 and 5 show the distribution of singular values of the two analysis dictionaries Ω_{AKSVD} and Ω_{Proposed} .

As observed, the ratio between the maximal and the minimal singular values of Ω_{Proposed} is almost the same as that of Ω , smaller than the one of Ω_{AKSVD} . This implies that our proposed algorithm, as expected, yields a dictionary closer to a TF than Ω_{AKSVD} owing to the penalty term $\mathcal{F}(\bar{\Omega})$.

FIGURE 3: Evolution of relative change of iterates $\|\Omega_k - \Omega_{k-1}\|_F / \|\Omega_{k-1}\|_F$ for $J = 50$, $N = 25$, and $L = 50000$.

FIGURE 4: Distribution of singular values of reference dictionary Ω , Ω_{AKSVD} , and Ω_{Proposed} for the noise-free case.

Performance Evaluation of the Obtained Dictionaries. The averaged representation error is defined as $\|Y - X_k\|_F / \sqrt{LN}$ for the dictionary Ω_k , where X_k is the estimate of the clean signals which are available for $\text{Alg}_{\text{AKSVD}}$ and can be obtained using the OBG for $\text{Alg}_{\text{Proposed}}$.

The performance of a given dictionary for denoising is usually measured with the averaged denoising error defined as $\|X_k - X\|_F / \sqrt{LN}$, where X is the true (clean) signals.

Figures 6 and 7 show comparison of the two algorithms in terms of the averaged representation error and the averaged denoising error, respectively. (In Figure 7, the plots for the

FIGURE 5: Distribution of singular values of reference dictionary Ω , Ω_{AKSVD} , and Ω_{Proposed} for the noisy case.

FIGURE 6: Evolution of the averaged representation error $\|Y - X_k\|_F / \sqrt{NL}$ versus iterations with $J = 50$, $N = 25$, and $L = 50000$.

noise-free case are not presented as they are the same as those shown in Figure 6 due to the fact that $Y = X$.)

Now, let us look at how close Ω_{Proposed} and Ω_{AKSVD} are to the reference dictionary Ω .

We say a row $\omega_m^\dagger$ in the true analysis dictionary Ω is recovered in $\hat{\Omega}$ if

$$\min_n \left(1 - \left| \hat{\omega}_n^\dagger \omega_m^{\dagger \mathcal{T}} \right| \right) < 0.01. \quad (40)$$

Denote J_k as the number of recovered rows in Ω_k . The percentage of the recovered rows denoted as $\rho_{\text{pr}}(k)$ in Ω_k is

FIGURE 7: Evolution of the average denoising error $\|X_k - X\|_F / \sqrt{LN}$ versus iterations with $J = 50$, $N = 25$, and $L = 50000$.

FIGURE 8: Evolution of the percentage of the recovered rows $\rho_{\text{pr}}(k)$ versus iterations with $J = 50$, $N = 25$, and $L = 50000$.

calculated with $\rho_{\text{pr}}(k) \triangleq J_k/J$. This is one of the performance indicators used in [12] for evaluating a learnt dictionary.

With signals generated above and the same setup as the one used in [12] except $N_{\text{ite}} = 200$ instead of 100, we run each of the $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$ and obtain a sequence of N_{ite} analysis dictionaries $\{\Omega_k\}$. Figure 8 shows the evolution of the percentage of the recovered rows for the two algorithms.

Comment 6. (i) It is seen that our proposed algorithm surprisingly outperforms $\text{Alg}_{\text{AKSVD}}$ in terms of averaged representation error, averaged denoising error, and percentage of recovered rows for both cases.

FIGURE 9: Evolution of the averaged representation error $\|Y - X_k\|_F / \sqrt{NL}$ versus iterations with $J = 40$, $N = 20$, and $L = 10000$.

FIGURE 11: Evolution of the percentage of the recovered rows $\rho_{\text{pr}}(k)$ versus iterations with $J = 40$, $N = 20$, and $L = 10000$.

FIGURE 10: Evolution of the average denoising error $\|X_k - X\|_F / \sqrt{LN}$ versus iterations with $J = 40$, $N = 20$, and $L = 10000$.

FIGURE 12: Relationship between the time (in seconds) consumed and the number of the training samples L .

(ii) According to equation (6) of [12], the averaged denoising error in the oracle setup (i.e., the true dictionary and all the cosupports are known) is given by $\sqrt{r/N}\sigma$, which is equal to 0.4σ for this example. As observed from Figure 7, the averaged denoising error for both algorithms seems to converge to a noise variance close to $\sigma = 0.04$. This indicates these two algorithms just yield one of the local minimal points of the problems though very close to the reference one.

Figures 9–11, as another example, display the results for the case of $N = 20$, $J = 40$, and $L = 10000$ and the vector

x_i living in a 3-dimensional subspace, leading to $\kappa = N - 3 = 17$. Here, we set $\lambda = 200$ and $\nu = 0.01$ for our proposed algorithm. The results are self-explanatory.

What is the most significant improvement of our proposed algorithm over $\text{Alg}_{\text{AKSVD}}$ is the implementation efficiency. This can be seen from Figure 12 which yields the relationship between the time (in seconds) used and the number of training samples L for the setting: $N = 25$, $J = 50$, and $N_{\text{ite}} = 200$. As an example, one notes that, for $L = 50000$, $\text{Alg}_{\text{AKSVD}}$ requires about 10^5 seconds, while our proposed algorithm needs less than 3×10^2 seconds.

TABLE 1: Denoising performance (image PSNR in dB) evaluated with five different pictures in different noise level σ for $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$.

σ	Algorithm	Lenna	Barbara	Boats	House	Peppers 256
5	$\text{Alg}_{\text{AKSVD}}$	37.32	36.88	36.60	38.04	36.89
	$\text{Alg}_{\text{Proposed}}$	37.54	37.11	36.65	38.11	36.98
10	$\text{Alg}_{\text{AKSVD}}$	33.35	32.00	32.07	33.89	32.11
	$\text{Alg}_{\text{Proposed}}$	33.46	32.85	32.07	33.88	32.29
15	$\text{Alg}_{\text{AKSVD}}$	31.05	29.22	29.62	31.35	29.53
	$\text{Alg}_{\text{Proposed}}$	31.30	29.73	29.91	31.60	29.94
20	$\text{Alg}_{\text{AKSVD}}$	29.45	27.30	27.92	29.58	27.79
	$\text{Alg}_{\text{Proposed}}$	29.49	28.11	28.54	29.64	28.18

4.2. *Experiments on Image Denoising.* Now, let us consider image denoising for five widely used images, “Lenna,” “Barbara,” “Boats,” “House,” and “Peppers.”

For a given noisy image with noise level $\sigma \in \{5, 10, 15, 20\}$ (which means each pixel is contaminated by a white Gaussian noise with i.i.d. zero-mean and σ^2 -variance), we generate a training set consisting of $L = 20,000$ overlapping 8×8 patch examples. Then, we run $\text{Alg}_{\text{Proposed}}$ and $\text{Alg}_{\text{AKSVD}}$ with $\lambda = 100$, $\nu = 0.01$, $r = 7$, $\kappa = 30$, and $N_{\text{ite}} = 20$ to learn the analysis dictionaries of size $J \times N = 128 \times 64$.

We apply the learnt dictionaries for patch-based image denoising. Each noised patch is denoised by using error-based OBG for sparse coding, which is the same as that in [12]. The denoising results with respect to image PSNR in dB are given in Table 1.

It is observed that denoising performance by $\text{Alg}_{\text{Proposed}}$ is in general better than that by $\text{Alg}_{\text{AKSVD}}$ in terms of PSNR.

Figure 13 shows the visual effects of denoising on the image “Barbara” with a noise level of $\sigma = 10$.

5. Conclusions

In this paper, we have investigated the problem of overcomplete transform learning for analysis model. An alternating minimization based iterative procedure has been proposed to solve the formulated optimal transform design problem. A closed-form solution has been derived for updating the transform. The superiority of the proposed approach in terms of the averaged representation and denoising errors, the percentage of the recovered rows, and, more significantly, the computation efficiency has been demonstrated with a series of experiments using both synthetic and real data.

As observed, both the AKSVD and our proposed algorithm demonstrate good convergence to a local minimum. Deeper studies along those lines [27, 28] are needed for theoretical guarantees of convergence to a global minimum and developing high performance algorithms for sparsifying dictionary and transform design. Furthermore, it is expected that the performance of the analysis KSVD algorithms in [12] can be improved with the proposed approach for updating dictionary embedded. Investigations along these lines are ongoing.

Appendix

Proof of Theorem 3

According to Laplace’s formula,

$$\det(Q) = \sum_{j=1}^N q_{ij} (-1)^{i+j} M_{ij}, \quad \forall i \in \{1, 2, \dots, N\}, \quad (\text{A.1})$$

where M_{ij} is the (i, j) th minor of matrix Q , that is, the determinant of $(N-1) \times (N-1)$, matrix that results from Q by removing the i th row and the j th column, while $(-1)^{i+j} M_{ij}$ is known as the (i, j) th cofactor of Q . Clearly, $(\partial \det(Q))/\partial q_{kl} = (-1)^{k+l} M_{kl}$ and hence

$$\frac{\partial \log \det(Q)}{\partial q_{kl}} = \frac{(-1)^{k+l} M_{kl}}{\det(Q)}, \quad \forall k, l \quad (\text{A.2})$$

as long as $\det(Q) > 0$.

Therefore, the gradient of $f(Q) \triangleq \log \det(Q)$ with respect to Q is given by

$$\nabla_Q f(Q) = \frac{\bar{Q}^\mathcal{T}}{\det(Q)} = Q^{-\mathcal{T}}, \quad (\text{A.3})$$

where $\bar{Q}$ is the adjugate matrix of Q , whose (i, j) th element is given by the (j, i) th cofactor of Q and satisfies $Q\bar{Q} = \det(Q)I_N$.

It can then be shown that for $g(\tilde{\Omega}) \triangleq f(Q)$ with $Q = \tilde{\Omega}^\mathcal{T} \tilde{\Omega}$

$$\begin{aligned} \nabla_{\tilde{\Omega}} g(\tilde{\Omega}) &= \tilde{\Omega} \left[\nabla_Q f(Q) + (\nabla_Q f(Q))^\mathcal{T} \right] \\ &= 2\tilde{\Omega} \left(\tilde{\Omega}^\mathcal{T} \tilde{\Omega} \right)^{-1} \end{aligned} \quad (\text{A.4})$$

and hence

$$\frac{d\mathcal{L}_1(\tilde{\Omega})}{d\tilde{\Omega}} = -2M + 2\tilde{\Omega}Z - 2\lambda\tilde{\Omega} \left(\tilde{\Omega}^\mathcal{T} \tilde{\Omega} \right)^{-1}. \quad (\text{A.5})$$

The optimal Ω should satisfy $(d\mathcal{L}_1(\tilde{\Omega})/d\tilde{\Omega})|_{\tilde{\Omega}=\Omega} = 0$, which leads to $\Omega Z \Omega^\mathcal{T} \Omega - M \Omega^\mathcal{T} \Omega - \lambda \Omega = \mathbf{0}$. Equivalently,

$$[WW^\mathcal{T} - \tilde{M}W^\mathcal{T} - \lambda I_J]W = \mathbf{0}, \quad (\text{A.6})$$

where $\tilde{M} \triangleq MZ_{\text{sqr}}^{-1}$ and $W \triangleq \Omega Z_{\text{sqr}}$ with Z_{sqr} the square root matrix of Z defined before.

Let

$$W = U_w \begin{bmatrix} \Sigma^2 \\ \mathbf{0} \end{bmatrix} V_w^\mathcal{T}, \quad (\text{A.7})$$

$$\tilde{M} = U_0 \begin{bmatrix} \Pi^2 \\ \mathbf{0} \end{bmatrix} V_0^\mathcal{T}$$

be an SVD of W and $\tilde{M}$, respectively. The above equation becomes

$$\begin{aligned} &\left\{ \begin{bmatrix} \Sigma^4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} - U_w^\mathcal{T} U_0 \begin{bmatrix} \Pi^2 \\ \mathbf{0} \end{bmatrix} V_0^\mathcal{T} V_w [\Sigma^2 \ \mathbf{0}] - \lambda I_J \right\} \\ &\cdot \begin{bmatrix} \Sigma^2 \\ \mathbf{0} \end{bmatrix} = \mathbf{0}. \end{aligned} \quad (\text{A.8})$$

FIGURE 13: “Barbara.” (a) The original; (b) noisy image with noise $\sigma = 10$ (PSNR = 28.13 dB); (c) denoised image using Ω_{AKSVD} (PSNR = 32.00 dB); (d) denoised image using Ω_{Proposed} (PSNR = 32.85 dB).

A solution (U_w, V_w, Σ^2) to the above is

$$\begin{aligned} V_w &= V_0, \\ U_w &= U_0, \\ \Sigma^2 &= \text{diag}(\sigma_1^2, \dots, \sigma_k^2, \dots, \sigma_N^2) \end{aligned} \quad (\text{A.9})$$

with $\sigma_k^2 > 0$ constrained by

$$\sigma_k^4 - \pi_k^2 \sigma_k^2 - \lambda = 0, \quad \forall k \quad (\text{A.10})$$

which yields (33) and hence (32) follows. This completes the proof.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by NSFC Grants 61273195, 61304124, 61473262, and 61503339.

References

- [1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, New York, NY, USA, 2010.
- [2] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [3] I. Tošić and P. Frossard, “Dictionary learning: what is the right representation for my signal?” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.
- [4] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [5] 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.
- [6] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [7] B. Ophir, M. Elad, N. Bertin, and M. D. Plumbley, “Sequential minimal eigenvalues—an approach to analysis dictionary learning,” in *Proceedings of the 19th European Signal Processing Conference (EUSIPCO '11)*, pp. 1465–1469, Barcelona, Spain, September 2011.

- [8] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," in *Proceedings of the 19th European Signal Processing Conference (EUSIPCO '11)*, pp. 1470–1474, Barcelona, Spain, September 2011.
- [9] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Noise aware analysis operator learning for approximately cosparse signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 5409–5412, March 2012.
- [10] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2341–2355, 2013.
- [11] R. Rubinstein, T. Faktor, and M. Elad, "K-SVD dictionary-learning for the analysis sparse model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 5405–5408, Kyoto, Japan, March 2012.
- [12] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: a dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.
- [13] K. Egan, S. O. Aase, and J. H. Hakon-Husoy, "Method of optimal directions for frame design," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, vol. 5, pp. 2443–2446, Phoenix, Ariz, USA, March 1999.
- [14] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [15] L. Zelnik-Manor, K. Rosenblum, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2386–2395, 2012.
- [16] J. A. Tropp, I. S. Dhillon, J. Heath, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.
- [17] I. S. Dhillon, J. Heath, T. Strohmer, and J. A. Tropp, "Constructing packings in Grassmannian manifolds via alternating projection," *Experimental Mathematics*, vol. 17, no. 1, pp. 9–35, 2008.
- [18] M. Yaghoobi, L. Daudet, and M. E. Davies, "Parametric dictionary design for sparse coding," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4800–4810, 2009.
- [19] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [20] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.
- [21] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.
- [22] S. Ravishanker and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, 2013.
- [23] S. Ravishanker and Y. Bresler, "Closed-form solutions within sparsifying transform learning," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 5378–5382, Vancouver, Canada, May 2013.
- [24] S. Ravishanker and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 3088–3092, Vancouver, Canada, May 2013.
- [25] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 89–105, 2014.
- [26] J. A. Fill and S. Janson, "Quicksort asymptotics," *Journal of Algorithms*, vol. 44, no. 1, pp. 4–28, 2002.
- [27] K. Schnass, "On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD," *Applied and Computational Harmonic Analysis*, vol. 37, no. 3, pp. 464–491, 2014.
- [28] R. Gribonval and K. Schnass, "Dictionary identification—sparse matrix-factorization via l_1 -minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.