Proximal Alternating Minimization for Analysis Dictionary Learning and Convergence Analysis

Zhenni Li, Shuxue Ding, Wuhui Chen, Zuyuan Yang, *Member, IEEE*, and Shengli Xie, *Senior Member, IEEE*

***Abstract*—The sparse analysis model is an alternative approach to the sparse synthesis model that has emerged recently. Most analysis dictionary learning problems based on the sparse analysis model require solving a class of challenging nonsmooth and even nonconvex optimization problems. Despite the fact that many numerical methods have been developed for solving these problems, it remains an open problem to find a numerical method that is not only empirically fast, but also has mathematically guaranteed strong convergence. In this paper, to promote stronger sparsity in solutions than the ℓ_1-norm, we employ the nonsmooth and nonconvex $\ell_{1/2}$-norm as a regularizer, and then we propose to use the proximal alternating minimization scheme for solving the nonconvex and nonsmooth problem, leading to an efficient and fast algorithm. More importantly, a rigorous convergence analysis shows that the proposed method satisfies the global convergence property: The whole sequence of iterates is convergent and converges to a critical point. The proposed algorithm has two stages: the analysis sparse-coding stage and the analysis dictionary-update stage. Besides the theoretical soundness, the practical benefit of the proposed method is validated by the synthetic and real-world data. Experiments show that the proposed method achieves better results with faster convergence compared to the state-of-the-art algorithms.**

***Index Terms*—Sparse analysis model, analysis dictionary learning, $\ell_{1/2}$ norm regularizer, convergence analysis.**

I. Introduction

SPARSITY-INSPIRED models have been proven to be valuable in signal processing, machine learning and artificial intelligence [1]–[5]. The best-known model is the sparse synthesis model [6], [7], where a natural signal is assumed to be created as a linear combination of a few atoms from an overcomplete dictionary. In the sparse synthesis model, only a small number of atoms are selected to represent each signal; therefore, every atom has the great significance. The wrong choice for the large weights of atoms leads to further deviation from the desired description. The sparse analysis model is an alternative approach to the sparse synthesis model that has emerged recently [8]. This model employs an overcomplete analysis dictionary $\mathbf{\Omega} \in \mathbb{R}^{r\times m}$ $(r > m)$ to "analyze" the signal $\mathbf{y}$ as $\mathbf{\Omega y} \approx \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^r$ is the analysis representation of $\mathbf{y} \in \mathbb{R}^m$ and contains l zero elements. l is defined as the cosparsity. The rows in $\mathbf{\Omega}$ represent dictionary atoms. The signal $\mathbf{y}$ is analyzed using all the rows of $\mathbf{\Omega}$. Furthermore, all atoms contribute equally to describing the signal, leading to a stabilizing effect in the recovery process.

The quality of reconstruction methods based on these two sparsity models strongly depends on the right choice of a suitable dictionary. An adaptive dictionary can provide comparable or better signal reconstruction quality in applications [9], [10], leading to the issues of dictionary learning. Learning algorithms aim at finding such an optimal dictionary by minimizing the average sparsity over a representative set of the signal. Analysis dictionary learning (ADL) based on the sparse analysis model has recently attracted increasing attention [11]–[19], but the analysis model has been studied less frequently than the well-known synthesis model. Several ADL algorithms have been proposed in the literature. Most use the ℓ_0 norm for strong sparsity. Unfortunately, exactly solving the ℓ_0 norm problem leads to a combinatorial optimization problem and its complexity is of NP-hard. Therefore, some approximate methods for solving the ℓ_0 norm problem have been developed, such as the analysis K-singular value decomposition (SVD) (AKSVD) algorithm [11], the transform K-SVD (TKSVD) algorithm [14], the analysis simultaneous codeword-optimization (ASimCO) [15] algorithm, and the learning overcomplete sparsifying transforms (LOST) [12]. The learning formulation of AKSVD involves NP-hard sparse coding. It applies the approximation method; i.e., the backward greedy algorithm to detect the sparsity, which tends to be computationally expensive in practice for large-scale problems. While TKSVD, ASimCO, and LOST employ another greedy algorithm, the hard thresholding operation [20], to detect the sparsity. These algorithms have low computational cost, but these algorithms yield only approximate results in many sparse signal estimation and reconstruction problems. In particular, these popular nonconvex algorithms do not have convergence guarantees. Some ADL algorithms use the convex ℓ_1 norm for sparsity instead of the ℓ_0 norm, and the corresponding optimization can be solved easily and efficiently, such as the analysis

Manuscript received July 12, 2017; revised December 1, 2017; accepted January 11, 2018. Date of publication March 2, 2018; date of current version November 21, 2018. This work was supported in part by the Guangdong Program for Support of Top-Notch Young Professionals under Grant 2014TQ01X144, in part by the National Natural Science Foundation of China under Grants 61722304 and 61333013, in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201610010196, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2014A030306037, and in part by the Science and Technology Plan Project of Guangdong under Grants 2014B090907010, 2015B010131014, and 2017B010125002. *(Corresponding author: Shengli Xie.)*

Z. Li, Z. Yang, and S. Xie are with the Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: lizhenni2012@gmail.com; yangzuyuan@aliyun.com; shlxie@gdut.edu.cn).

S. Ding is with the Faculty of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu-shi 965-8580, Japan (e-mail: sding@u-aizu.ac.jp).

W. Chen is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China (e-mail: chenwuhui21@gmail.com).

Digital Object Identifier 10.1109/TETCI.2018.2806890

operator learning (AOL) algorithm [13], the orthogonality-constrained ADL with iterative hard thresholding (OIHT-ADL) [18] algorithm, and the fast proximal analysis dictionary learning (FPADL) algorithm [19]. The corresponding optimization is convex and easy to solve. However, the ℓ_1 norm may not induce adequate sparsity in some applications [21], [22]. Furthermore, the convex models based on the ℓ_1 norm have been shown to be suboptimal in many cases [23], [24]. Indeed, the ℓ_1 norm often leads to overpenalization of large elements of a sparse vector, because it is a loose approximation of the ℓ_0 norm [25].

In this paper, we aim to propose a efficient, fast algorithm and establish the mathematically guaranteed strong convergence for nonconvex and nonsmooth ADL problem. Inspired by the recent development of the $\ell_{1/2}$ norm regularizer theory in synthesis model [21], we propose to apply the nonconvex and nonsmooth $\ell_{1/2}$ norm to ADL as a regularizer to obtain the strong sparsity-promoting solutions. The $\ell_{1/2}$ norm is closer to the ℓ_0 norm and can obtain sparser solutions than the ℓ_1 norm. However, the $\ell_{1/2}$ norm regularizer results in a complex nonconvex optimization problem that is harder to solve efficiently than the convex optimization problem. In particular, the convergence analysis of a nonconvex algorithm is in general tricky. We propose to use a fast proximal alternating minimization scheme [26], [27] to solve the nonconvex and nonsmooth ADL problem arising from the $\ell_{1/2}$ norm. The proximal alternating minimization algorithm is fast and efficient, and can solve a broad class of nonconvex and nonsmooth minimization problems. The proposed algorithm is called PADL-$\ell_{1/2}$ which has two stages: the analysis sparse-coding stage and the analysis dictionary-update stage. The proposed algorithm not only should be computationally efficient in practice but also should have strong convergence guaranteed by theoretical analysis; e.g., the global convergence property: the whole sequence generated by the method converges to a critical point of the problem. Our preliminary results of using the $\ell_{1/2}$ norm as the regularizer were presented in [28]. We propose to depart from [28] and employ the proximal alternating minimization scheme to solve the nonconvex ADL problem and especially to establish the global convergence analysis. To the best of our knowledge, no theoretical or empirical global convergence properties have been demonstrated for the ADL problem with the $\ell_{1/2}$ norm regularizer.

The rest of this paper is organized as follows. The conventional ADL problem is reviewed in Section II. We formulate our ADL problem and develop a novel and efficient ADL algorithm in Section III. The adaptive parameter settings are also described in Section III. The theoretical findings are verified by experiments in Section IV. Finally, conclusions are drawn in Section V.

II. Preliminaries

In this section, we first introduce some notation. Then we introduce the conventional ADL formulation.

A. Notation

We briefly summarize notation in this paper. A boldface uppercase letter such as $\mathbf{X}$ denotes a matrix, and $\mathbf{x}_l$ denotes the l-th column vector in $\mathbf{X}$. A lowercase element x_{jl} denotes the entry in the j-th row and the l-th column of $\mathbf{X}$. A lowercase element x_j denotes the j-th entry in the vector $\mathbf{x}$. A boldface uppercase Greek letter $\mathbf{\Omega}$ denotes an analysis dictionary matrix and $\mathbf{w}^{(i)}$ is the i-th row vector in $\mathbf{\Omega}$, i.e., the i-th atom. The Frobenius norm of $\mathbf{X}$ is defined as $\| \mathbf{X} \|_F = (\sum_{l,j} \mid x_{jl} \mid^2)^{1/2}$ and ℓ_2 norm of $\mathbf{x}$ is defined as $\| \mathbf{x} \|_2 = (\sum_j \mid x_j \mid^2)^{1/2}$. The ℓ_1 norm of $\mathbf{X}$ and $\mathbf{x}$ are defined as $\| \mathbf{X} \|_1 = \sum_{j,l} \mid x_{jl} \mid$ and $\| \mathbf{x} \|_1 = \sum_j \mid x_j \mid$, respectively. The $\ell_{1/2}$ norm of $\mathbf{X}$ and $\mathbf{x}$ are defined as $\| \mathbf{X} \|_{1/2} = (\sum_{l,j} \mid x_{jl} \mid^{1/2})^2$ and $\| \mathbf{x} \|_{1/2} = (\sum_j \mid x_j \mid^{1/2})^2$, respectively. $\mathbf{I}$ is the identity operator. In this paper, all parameters take real values.

B. Analysis Dictionary Learning (ADL)

Given a matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ of observed signals, the aim of ADL in the analysis model is to find the analysis dictionary $\mathbf{\Omega} \in \mathbb{R}^{r \times m}$ with $m < r$ such that $\mathbf{\Omega Y}$ is as sparse as possible. This strategy can be formally expressed as the following optimization problem:

$$\min_{\mathbf{\Omega}, \mathbf{X}} \frac{1}{2} \|\mathbf{\Omega Y} - \mathbf{X}\|_F^2 + \lambda Sp(\mathbf{X}). \tag{1}$$

where λ is the regularization parameter and $Sp(\cdot)$ is a sparsity-promoting function. In general, the sparsity measures ℓ_0 norm and ℓ_1 norm are used for this purpose. This analysis model for ADL is also named as the transform model [29], where $\mathbf{X} \in \mathbb{R}^{r \times n}$ as the analysis representation matrix and an approximation error $\|\mathbf{\Omega Y} - \mathbf{X}\|_F^2$ are introduced. In the transform model, ADL has two stages: the analysis sparse-coding stage (finding the sparsest $\mathbf{X}$) and the analysis dictionary-update stage (learning a better $\mathbf{\Omega}$).

However, unconstrained minimizations of (1), based on $\mathbf{\Omega}$, have some trivial solutions. One solution for such a minimization problem is $\mathbf{\Omega} = \mathbf{0}$!. To avoid such trivial solutions, the following constraints on $\mathbf{\Omega}$ could be imposed.

(i) Row-norm constraints: $\| \mathbf{w}^{(i)} \|_2 = c$, $\forall i \in [1, r]$.

(ii) Full-column rank constraints: $rk(\mathbf{\Omega}) = m$.

(iii) No linearly dependent rows in $\mathbf{\Omega}$: $\mathbf{w}^{(i)} \neq \pm \mathbf{w}^{(j)}, i \neq j$.

Recent work has found that the above constraints are insufficient for the ADL task, so the uniform normalized tight frame (UNTF) constraint [13] attracts our interest, particularly because it offers better properties and greater efficiency. The UNTF constraint is a combination of the row-norm constraint and the full-column rank constraint; i.e., an orthonormality constraint $\mathbf{\Omega}^T \mathbf{\Omega} = \mathbf{I}$, which can be written as:

$$\mathcal{L} = \{\mathbf{\Omega} : \mathbf{\Omega}^T \mathbf{\Omega} = \mathbf{I}, \text{and } \| \mathbf{w}^{(i)} \|_2 = c, \forall i\}. \tag{2}$$

The UNTF set not only avoids the trivial solutions but also enhances the incoherence between the dictionary atoms, so that a sparse representation with such a dictionary is more robust than that using a dictionary without the UNTF constraint. Although it cannot be claimed that the UNTF constraint is the optimal choice, it has been shown that an analysis dictionary constrained to the UNTF set avoids degenerate solutions; moreover, it has low mutual coherence between the dictionary atoms.

III. Proposed Algorithm

In this section, we first formulate our ADL problem where the $\ell_{1/2}$ norm is used for strong sparsity and the UNTF constraint is used to avoid the trivial solutions. Second, we propose to use the proximal alternating minimization scheme to solve the nonconvex problem based on two stages: the analysis sparse-coding stage and the analysis dictionary-update stage. In the analysis sparse-coding stage, we solve the nonconvex and nonsmooth $\ell_{1/2}$ norm regularized problem by optimizing a number of one-dimensional minimization problems and obtain the proximal operator for the $\ell_{1/2}$ norm. In the third part, we summarize the proposed algorithm. In addition, setting adaptive parameters are discussed in the last part.

A. The Learning Goal

We begin to describe our ADL problem based on the transform model (1), where $G(\mathbf{X})$ is the $\ell_{1/2}$ norm over the analysis representation $\mathbf{X} \in \mathbb{R}^{r\times n}$ as sparsity regularizer and the UNTF constraint—i.e., $\mathcal{L} = \{\mathbf{\Omega} : \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}, \text{and } \| \mathbf{w}^{(i)} \|_2 = c, \forall i \in [1, r]\}$—is imposed on the dictionary $\mathbf{\Omega} \in \mathbb{R}^{r\times m}$, where $c = \sqrt{\frac{m}{r}}$. Hence, our ADL problem can be formulated as the minimization problem,

$$\min_{\mathbf{\Omega},\mathbf{X}} \phi(\mathbf{\Omega}, \mathbf{X}) = F(\mathbf{X}) + Q(\mathbf{\Omega}, \mathbf{X}) + G(\mathbf{\Omega}), \tag{3}$$

where

$$Q(\mathbf{\Omega}, \mathbf{X}) = \frac{1}{2}\|\mathbf{\Omega}\mathbf{Y} - \mathbf{X}\|_F^2, \tag{4}$$

$$G = \delta_{\mathcal{L}}(\mathbf{\Omega}), \tag{5}$$

$$F(\mathbf{X}) = \lambda \ \| \mathbf{X} \|_{1/2}^{1/2}, \tag{6}$$

here a regularization parameter λ controls the sparsity of $\mathbf{X}$. $F(\mathbf{X})$ is lower semi-continuous and even nonconvex function. $G(\mathbf{\Omega})$ is an indicator function. $Q(\mathbf{\Omega}, \mathbf{X})$ is the C^1 function[1] and $\bigtriangledown Q$ is Lipschitz continuous on bounded subsets of $\mathbb{R}$. Compared with the ℓ_1 norm, the nonconvex $\ell_{1/2}$ norm regularizer can usually induce better sparsity, but the corresponding nonconvex regularized optimization problems are generally more difficult to solve and the convergence analysis of a nonconvex algorithm is in general tricky. In the following part, we will propose the proximal alternating minimization framework for solving our nonconvex ADL problem (3) with the global convergence property.

B. The Proximal Alternating Minimization Scheme for ADL With the $\ell_{1/2}$-Norm Regularizer

We propose to solve (3) based on the proximal alternating minimization scheme [26], [27] which is about tacking the nonconvex and nonsmooth minimization problem of the form,

$$\min_{\mathbf{x},\mathbf{y}} H(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) + Q(\mathbf{x}, \mathbf{y}) + G(\mathbf{y}), \tag{7}$$

where $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $G : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are lower semi-continuous, possibly nonconvex functions. $Q : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the C^1 function and $\bigtriangledown Q$ is Lipschitz continuous on bounded subsets of $\mathbb{R}^n \times \mathbb{R}^m$. The proximal alternating minimization algorithm updates the $(\mathbf{x}, \mathbf{y})$ via as follows,

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} F(\mathbf{x}) + \hat{Q}(\mathbf{x}) + \frac{\mu^k}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2, \tag{8}$$

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} \hat{Q}(\mathbf{y}) + G(\mathbf{y}) + \frac{\nu^k}{2}\|\mathbf{y} - \mathbf{y}^k\|_2^2, \tag{9}$$

where $\hat{Q}(\mathbf{x}) = Q(\mathbf{x}^k, \mathbf{y}^k) + \langle \mathbf{x} - \mathbf{x}^k, \bigtriangledown_{\mathbf{x}} Q(\mathbf{x}^k, \mathbf{y}^k)\rangle$ and $\hat{Q}(\mathbf{y}) = Q(\mathbf{x}^{k+1}, \mathbf{y}^k) + \langle \mathbf{y} - \mathbf{y}^k, \bigtriangledown_{\mathbf{y}} Q(\mathbf{x}^{k+1}, \mathbf{y}^k)\rangle$. μ^k and ν^k are two appropriately chosen step sizes. The iterations (8) and (9) can be rewritten as,

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{\mu^k}{2} \left\| \mathbf{x} - \left(\mathbf{x}^k - \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}} Q(\mathbf{x}^k, \mathbf{y}^k) \right) \right\|_2^2 + F(\mathbf{x}), \tag{10}$$

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} \frac{\nu^k}{2} \left\| \mathbf{y} - \left(\mathbf{y}^k - \frac{1}{\nu^k} \bigtriangledown_{\mathbf{y}} Q\left(\mathbf{x}^{k+1}, \mathbf{y}^k\right) \right) \right\|_2^2 + G(\mathbf{y}), \tag{11}$$

The iterations (10) and (11) can be rewritten by using the proximal operator [30]. Then, the update rule of (10) and (11) can be rewritten as,

$$\mathbf{x}^{k+1} = \mathrm{Prox}_{\mu^k, F}\left(\mathbf{x}^k - \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}} Q(\mathbf{x}^k, \mathbf{y}^k) \right), \tag{12}$$

$$\mathbf{y}^{k+1} = \mathrm{Prox}_{\nu^k, G}\left(\mathbf{y}^k - \frac{1}{\nu^k} \bigtriangledown_{\mathbf{y}} Q(\mathbf{x}^{k+1}, \mathbf{y}^k) \right). \tag{13}$$

When F or G is indicator function δ of a nonempty and closed set $\mathcal{C}$, i.e., for the function $\delta_{\mathcal{C}}$. For all $x \in \mathbb{R}^n$, we set,

$$\delta_{\mathcal{C}}(x) = \begin{cases} 0 \quad , & \text{if} \quad x \in \mathcal{C}; \\ +\infty \, , & \text{otherwise}, \end{cases} \tag{14}$$

The proximal map reduces to the projection operator on $\mathcal{C}$, defined by

$$\mathrm{P}_{\mathcal{C}}(\mathbf{y}) := \arg\min_{\mathbf{x}\in\mathcal{C}} \ \| \mathbf{x} - \mathbf{y} \|_2^2, \tag{15}$$

$\mathrm{P}_{\mathcal{C}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is projection mapping onto $\mathcal{C}$.

Applying the proximal alternating minimization scheme to (3), our proposed algorithm consists two stages: analysis sparse coding and analysis dictionary update in the following.

Analysis sparse coding: The purpose of the analysis sparse stage is to obtain the analysis representations $\mathbf{X}$ of the observed signals $\mathbf{Y}$ based on a given dictionary $\mathbf{\Omega}$. Hence, we solve (3) with respect to $\mathbf{X}$ as follows:

$$\min_{\mathbf{X}} \phi(\mathbf{X}) = F(\mathbf{X}) + Q(\mathbf{\Omega}, \mathbf{X}). \tag{16}$$

[1] C^1: first derivatives are continuous.

Consider that $\phi(\mathbf{X})$ is separable according to the columns of $\mathbf{X}$; that is,

$$\phi(\mathbf{X}) = \sum_{l=1}^{n} \left(F(\mathbf{x}_l) + Q(\mathbf{\Omega}, \mathbf{x}_l)\right),$$
$$\text{with} \quad Q(\mathbf{\Omega}, \mathbf{x}_l) = \frac{1}{2}\|\mathbf{\Omega}\mathbf{y}_l - \mathbf{x}_l\|_2^2,$$
$$\text{and} \quad F(\mathbf{x}_l) = \lambda_l \parallel \mathbf{x}_l \parallel_{1/2}^{1/2}, \tag{17}$$

where $\mathbf{y}_l$ and $\mathbf{x}_l$ are the l-th columns of $\mathbf{Y}$ and $\mathbf{X}$ for $\forall l = 1, ..., n$. Because the summation of (17) is separable, applying the proximal minimization algorithm to the problem (17), the iteration of $\mathbf{x}_l$ is obtained as,

$$\mathbf{x}_l^{k+1} = \arg\min_{\mathbf{x}_l} F(\mathbf{x}_l) + \hat{Q}(\mathbf{x}_l) + \frac{\mu^k}{2}\|\mathbf{x}_l - \mathbf{x}_l^k\|_2^2, \forall l \tag{18}$$

where $\hat{Q}(\mathbf{x}_l) = Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) + \langle \mathbf{x}_l - \mathbf{x}_l^k, \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k)\rangle$ and $\bigtriangledown_{\mathbf{x}_l} Q = \mathbf{x}_l - \mathbf{\Omega}\mathbf{y}_l$. And the above iteration can be rewritten as

$$\mathbf{x}_l^{k+1} = \arg\min_{\mathbf{x}_l} \frac{\mu^k}{2}\|\mathbf{x}_l - \mathbf{a}_l\|_2^2 + \lambda_l \parallel \mathbf{x}_l \parallel_{1/2}^{1/2} \quad \forall l. \tag{19}$$

where $\mathbf{a}_l = \mathbf{x}_l^k - \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k)$. Inspired by the proximal operator for ℓ_1 norm regularization [30], we try to derive the proximal operator for the $\ell_{1/2}$ norm in (19) which can be transformed to a number of one-dimensional problems as follows,

$$\min_{x_{jl}} \sum_j \frac{\mu_k}{2}(x_{jl} - a_{jl})^2 + \lambda_l |x_{jl}|^{1/2} \quad \forall l. \tag{20}$$

The summation is separable so that the problem (20) can be solved by minimizing a number of one-dimensional problems.

Considering $x_{jl} \neq 0$, we take the derivative of (20) to obtain the optimal solution. Because $\nabla(|x_{jl}|^{1/2}) = \frac{\text{sign}(x_{jl})}{2\sqrt{|x_{jl}|}}$, we have

$$x_{jl} - a_{jl} + \frac{\lambda_l}{\mu_k} \frac{\text{sign}(x_{jl})}{2\sqrt{|x_{jl}|}} = 0, \tag{21}$$

for any fixed j, the solution x_{jl}^* satisfies $x_{jl}^* a_{jl} > 0$.

When $x_{jl} > 0$, we denote $\sqrt{|x_{jl}|} = z$ and have $x_{jl} = z^2$ and then (21) can be written as

$$z^3 - a_{jl} z + \frac{\lambda}{2} = 0, \tag{22}$$

Because (22) is a cubic equation, we can obtain the solutions by the Cardano formula [31]. Denoting $r = \sqrt{|a_{jl}|/3}$, $p = -a_{jl}$ and $q = \lambda_l/2\mu_k$. When $\frac{q^2}{4} + \frac{p^3}{27} < 0$—i.e., $a_{jl} > \frac{3}{4}(\frac{2\lambda_l}{\mu_k})^{2/3}$—(22) has three roots. Denoting $\varphi_\lambda = \arccos(\frac{q}{2r^3})$, the three roots of (22) can be given as $z_1 = -2r\cos(\varphi_\lambda/3)$, $z_2 = 2r\cos(\pi/3 + \varphi_\lambda/3)$ and $z_3 = 2r\cos(\pi/3 - \varphi_\lambda/3)$. Because $x_{jl} > 0$, z_1 is not the solution of (22). Then we further check z_2 and z_3. $z_3 > z_2$ and thus z_3 is the unique solution of (22). Therefore, in this case, (22) has a unique solution that is given as $x_{jl} = \frac{2}{3}a_{jl}(1 + \cos(\frac{2\pi}{3} - \frac{2\varphi_\lambda(a_{jl})}{3}))$, where $\varphi_\lambda(a_{jl}) = \arccos(\frac{\lambda_l}{4\mu_k}(\frac{|a_{jl}|}{3})^{-\frac{3}{2}})$.

When $x_{jl} < 0$, we denote $\sqrt{|x_{jl}|} = z$ and have $x_{jl} = -z^2$ and then (21) can be written as

$$z^3 + a_{jl} z + \frac{\lambda}{2} = 0. \tag{23}$$

Similar with the case of $x_{jl} > 0$, we obtain the unique solution written as $x_{jl} = -\frac{2}{3}a_{lj}(1 + \cos(\frac{2\pi}{3} - \frac{2\varphi_\lambda(a_{jl})}{3}))$, when $a_{jl} < -\frac{3}{4}(\frac{2\lambda_l}{\mu_k})^{2/3}$.

When $x_{jl} = 0 \Leftrightarrow | a_{jl} | \leq \frac{3}{4}(\frac{2\lambda_l}{\mu_k})^{2/3}$.

We summarize all the solutions of one-dimensional problems with respect to x_{jl} and thus obtain the proximal operator for the $\ell_{1/2}$ norm regularizer,

$$x_{jl}^* = [\text{Prox}_{\lambda_l \mu_k, 1/2}(\mathbf{a}_l)]_j = \begin{cases} f_{\lambda_l, 1/2}(a_{jl}), & | a_{jl} | > y_l^*; \\ 0, & \text{otherwise}, \end{cases} \tag{24}$$

where $f_{\lambda_l, 1/2}(a_{jl}) = \frac{2}{3}\text{sign}(a_{jl})|a_{jl}|(1 + \cos(\frac{2\pi}{3} - \frac{2\varphi_\lambda(a_{jl})}{3}))$ and $\varphi_\lambda(a_{jl}) = \arccos(\frac{\lambda_l}{4\mu_k}(\frac{|a_{jl}|}{3})^{-\frac{3}{2}})$. $y_l^* = \frac{3}{4}(\frac{2\lambda_l}{\mu_k})^{2/3}$ is called the thresholding value. The symbol $\text{Prox}_{\lambda_l \mu_k, 1/2}$ is the proximal operator for the $\ell_{1/2}$ norm.

Thus the iteration of $\mathbf{x}_l$ can be obtained exactly using the proximal operator $\text{Prox}_{\lambda_l \mu_k, 1/2}$ as follows,

$$\mathbf{x}_l^{k+1} = \text{Prox}_{\lambda_l \mu_k, 1/2}(\mathbf{a}_l) \tag{25}$$

By this, we can optimize $\mathbf{X}$ by the optimizing the set of $\{\mathbf{x}_l\}$ sequentially for $\forall l = 1, ..., n$.

Analysis dictionary update: The analysis dictionary update stage aims to update $\mathbf{\Omega}$. We solve (3) with respect to $\mathbf{\Omega}$ as follows:

$$\min_{\mathbf{\Omega}} \phi(\mathbf{\Omega}) = Q(\mathbf{\Omega}, \mathbf{X}) + G(\mathbf{\Omega}). \tag{26}$$

Applying the proximal minimization algorithm to the problem (26), the iteration of $\mathbf{\Omega}$ is given as

$$\mathbf{\Omega}^{k+1} = \arg\min_{\mathbf{\Omega}} \hat{Q}(\mathbf{\Omega}, \mathbf{X}^{k+1}) + G(\mathbf{\Omega}) + \frac{\nu^k}{2}\|\mathbf{\Omega} - \mathbf{\Omega}^k\|_F^2, \tag{27}$$

where $\hat{Q}(\mathbf{\Omega}) = Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) + \langle \mathbf{\Omega} - \mathbf{\Omega}^k, \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k)\rangle$. Then, we have

$$\mathbf{\Omega}^{k+1} = \arg\min_{\mathbf{\Omega}} \frac{\nu^k}{2} \left\| \mathbf{\Omega} - \left(\mathbf{\Omega}^k - \frac{1}{\nu^k} \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) \right) \right\|_2^2 + G(\mathbf{\Omega}), \tag{28}$$

where $G(\mathbf{\Omega})$ is an indicator function of the UNTF set. The uniform normalised (UN) set and tight frame (TF) set are closed sets [32]. Denote $\mathbf{B} = \mathbf{\Omega}^k - \frac{1}{\nu^k} \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k)$. According to the projection operator (15), the iteration on $\mathbf{\Omega}$ is,

$$\mathbf{\Omega}^{k+1} = \text{P}_{\mathcal{L}}(\mathbf{B}), \tag{29}$$

$\text{P}_{\mathcal{L}}$ is the projection on the set $\mathcal{L} = \{\mathbf{\Omega} : \mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}, \text{and} \parallel \mathbf{w}^{(i)} \parallel_2 = \sqrt{\frac{m}{r}}, \forall i\}$ which is a combination of row norm $\parallel \mathbf{w}^{(i)} \parallel_2 = \sqrt{\frac{m}{r}}, \forall i\}$ and full column rank constraint $\mathbf{\Omega}^T\mathbf{\Omega} = \mathbf{I}$.

P_{TF} is a projection of a full rank matrix onto the TF manifold and this can be realized by calculating a singular value

Algorithm 1: PADL-$\ell_{1/2}$.

Input: Data matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$.
1: Initialize $\mathbf{\Omega}_0 \in \mathbb{R}^{r\times m}$ and $\mathbf{X}_0 \in \mathbb{R}^{r\times n}$,
2: for $k = 1, 2, ...$ do
3: Adjust regularization parameter λ_l^k, μ^k and ν^k, $l = 1, ..., n$,
4: Compute the analysis representations $\mathbf{X}_k$ based on equation (25),
5: Update $\mathbf{\Omega}_k$ based on equation (32),
6: end for
output: $\mathbf{\Omega}$.

decomposition of the linear operator [32], which is defined as

$$\mathrm{P}_{TF}(\mathbf{B}) = \mathbf{U}\mathbf{I}_{r\times m}\mathbf{V}^T, \tag{30}$$

where $\mathbf{I}_{r\times m}$ is a diagonal matrix with identity on the main diagonal. The projection can enforce the full column rank and thus avoid the trivial solution well. However, it cannot avoid a subset of rows of $\mathbf{\Omega}$ to be possibly zeros. Hence, we need normalize the rows of $\mathbf{\Omega}$ to prevent the scaling ambiguity and replace the zero rows by the normalized random vectors.

P_{UN} is the projection of an operator with non-zero rows, which can be done by scaling each row to have norm $c = \sqrt{\frac{m}{r}}$. If a row is zero, we set the row to a normalized random vector. This projection is defined as,

$$\mathrm{P}_{UN}(\mathbf{\Omega}) = [\mathrm{P}_{UN}(\mathbf{w}^{(i)})]_i = \begin{cases} \sqrt{\frac{m}{r}}\frac{\mathbf{w}^{(i)}}{\|\mathbf{w}^{(i)}\|}, & \|\mathbf{w}^{(i)}\| \neq 0; \\ \mathbf{v}, & \text{otherwise}, \end{cases} \tag{31}$$

where $\mathbf{w}^{(i)}$ is i-th row in $\mathbf{\Omega}$ and $\mathbf{v}$ is a normalized random vector on the unit sphere.

Hence, $\mathrm{P}_{\mathcal{L}}$ relies on the projections, i.e., P_{TF} and P_{UN}. The iteration on $\mathbf{\Omega}$ can be rewritten as,

$$\mathbf{\Omega}^{k+1} = \mathrm{P}_{\mathcal{L}}(\mathbf{B}) = \mathrm{P}_{UN}\big(\mathrm{P}_{TF}(\mathbf{B})\big). \tag{32}$$

C. The Overall Algorithm

The proposed algorithm alternates between the analysis sparse-coding stage and the analysis dictionary-update stage for a number of iterations. The structure of the proposed algorithm is outlined in Algorithm 1.

D. Parameters Setting

Because our proposed algorithm includes two step sizes μ^k and ν^k, and a regularizer λ_l^k, which are needed to be set in the calculation.

The step sizes μ^k and ν^k are set according to the following criteria:

1) $\mu^k, \nu^k \in (e, f)$, where $0 < e < f$ are two constants.
2) $L_{\mathbf{x}} \leq \mu^k < f$ and $L_{\mathbf{\Omega}} \leq \nu^k < f$, where the Lipschitz continuous $L_{\mathbf{x}}$ and $L_{\mathbf{\Omega}}$ are set to be the maximum eigenvalue of the matrices $\| \nabla^2_{\mathbf{x}} Q(\mathbf{x}, \mathbf{\Omega}) \|_F$ and $\| \nabla^2_{\mathbf{\Omega}} Q(\mathbf{X}, \mathbf{\Omega}) \|_F$.

The minimization of the objective function (17) with respect to the analysis vector $\mathbf{x}_l \in \mathbb{R}^r$ $(l = 1, ..., n)$ is an s-sparse problem. Each $\mathbf{x}_l$ may have its own regularization parameter value, and so we may use n independent parameter values for each iteration. This is denoted by the use of λ_l^k in the k-th iteration. We then need to solve an $\ell_{1/2}$ norm regularizer problem that is restricted to the subregion $\Gamma_s = \{\mathbf{x}_l = (x_{1l}, x_{2l}, ..., x_{rl}) : supp(x) = s\}$ of $\mathbb{R}^r$, where x_{lr} is the r-th entry of $\mathbf{x}_l$ and $supp(x)$ is the support set of x, i.e., $supp(x) = s : x_s \neq 0$. A vector $\mathbf{x}_l$ is said to be s-sparse and it contains s elements. Assume that $| a_{1l} | \geq | a_{2l} | \geq \ ... \geq | a_{rl} |$ and the set $T_s = | a_{sl} |$. Then, according to (24), we have the inequalities

$$| a_{il} | \ > \frac{3}{4}\left(\frac{2\lambda_l}{\mu_k}\right)^{2/3} \Longleftrightarrow i \in 1, 2, ..., s,$$

$$|a_{jl}| \leq \frac{3}{4}\left(\frac{2\lambda_l}{\mu_k}\right)^{2/3} \Longleftrightarrow j \in s+1, s+2, ..., r. \tag{33}$$

These imply that

$$\lambda_l^k = \sqrt{\frac{16}{27}}\mu_k T_{s+1}^{\frac{3}{2}}. \tag{34}$$

In this way, each regularization parameter λ_l^k can be determined adaptively, which can evolve independently and can decrease monotonically with each iteration of the algorithm.

IV. Global Convergence of Algorithm 1

Before proving the global convergence property of Algorithm 1, we define critical points for nonconvex and nonsmooth functions.

Definition 1: Assume the nonconvex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper and lower semi-continuous function and $\mathrm{dom} f = \{x \in \mathbb{R}^n : f(x) < +\infty\}$.

- The Fr*é* chet subdifferential of f is defined as

$$\hat{\partial} f(x) = \left\{u : \lim_{y\to x} \inf_{y\neq x} \frac{(f(y) - f(x) - \langle u, y - x\rangle}{\|y - x\|} \geq 0\right\}, \tag{35}$$

  if $x \in \mathrm{dom} f$, and $\varnothing$ otherwise.
- The Limiting Subdifferential of f at x is defined as

$$\partial f(x) = \{u \in \mathbb{R} : \exists x^k \to x, f(x^k) \to f(x) \quad \text{and} \quad u^k \in \hat{\partial} f(x^k) \to u\}. \tag{36}$$

- x is a critical point of f if $0 \in \partial f(x)$.

Theorem 1 (Convergence property): The sequence $\{\mathbf{X}^k, \mathbf{\Omega}^k\}$ (k is the iteration number) generated by Algorithm 1 is bounded and converges to the critical points of the objective function (3).

Proof: See Appendix. ■

V. Simulations

In this section, to evaluate the performance of the proposed algorithm, we present the results of some numerical experiments. The first experiment uses synthetic data generated from a known dictionary taken as the reference dictionary. We apply the algorithms to the synthetic data to learn dictionaries for comparison

with the reference dictionary. The strong ability to recover the dictionary demonstrates the effectiveness of the algorithm. We also illustrate the convergence behavior of our proposed algorithm in different cases. Second, we experiment with real-world audio data to determine whether the dictionary learned by the proposed algorithm can represent the original signal as a sparse approximation. In this experiment, we verify the robustness of the $\ell_{1/2}$ norm for sparsity.

In the experiments, all programs were coded in Matlab and were run within Matlab (R2016a) on a PC with a 3.0 GHz Intel Core i5 CPU and 16 GB of memory, under the Microsoft Windows 7 operating system.

A. Experiments on Synthetic Data

When the reference dictionary is unknown, evaluating the performance of dictionary learning algorithms is a difficult task. Here we focus on synthetic data built from a known dictionary. We compare the dictionaries learned by the ADL algorithms with the reference dictionary, so that the strong ability to recover the dictionary can demonstrate the effectiveness of our algorithm.

1) Experimental Setup: First, the reference dictionary $\mathbf{\Omega} \in \mathbb{R}^{r\times 16}$ was generated by projecting a random $\mathbf{\Omega}_0 \in \mathbb{R}^{r\times 16}$ repeatedly onto the UN and TF sets. The $\mathbf{\Omega}_0$ matrix had an independent and identically distributed (IID) Gaussian distribution with zero mean and unit variance. Then, synthetic data for the sample set $\mathbf{Y} \in \mathbb{R}^{16\times n}$ comprising signals $\mathbf{y}_i$ $(i = 1, ..., n)$ were obtained from the reference dictionary $\mathbf{\Omega}$. Every signal $\mathbf{y}_i$ was generated as to have the cosparsity l with respect to the reference dictionary $\mathbf{\Omega}$. This is realized by randomly selecting l rows from $\mathbf{\Omega}$ and then again randomly generating a vector from the orthogonal complement space of these chosen rows. Such a vector $\mathbf{y}_i$ has l zero elements in $\mathbf{\Omega}\mathbf{y}_i$, and it therefore had l cosparsity. The initial dictionary for the algorithms was generated by projecting a matrix $\mathbf{\Omega}_{in} = \mathbf{\Omega} + \gamma\mathbf{N}$ repeatedly onto the UNTF set. Here, γ was a constant that determined the deviation of the initial dictionary from the reference dictionary, and $\mathbf{N}$ was a normalized random matrix. Larger values of γ indicate larger deviations of the initial dictionary from the reference dictionary. If $\gamma \to \infty$, the initial dictionary would be random.

2) Performance Criteria: We used three approaches to evaluate the performance of the algorithms in the experiments.

First, the recovery ratio for the reference dictionary rows was used to evaluate the performance of the algorithms. If the maximum ℓ_2 distance between any recovered atom (row) $\hat{\mathbf{w}}^{(i)}$ in the learned dictionary $\hat{\mathbf{\Omega}}$ and the closest atom $\mathbf{w}^{(j)}$ in the reference dictionary $\mathbf{\Omega}$ was less than 0.01, it was considered to be a successful recovery of one atom. Higher recovery ratios indicate better ADL algorithms.

Second, the recovery cosparsity of the original signals $\mathbf{Y}$ with respect to the learned dictionary $\hat{\mathbf{\Omega}}$ was used as another performance measure [15]. The goal of ADL is to obtain an analysis dictionary for which the analysis representations of the signals are sparse, and thus the recovery cosparsity of the original signals is very significant. $\| \mathbf{b} \|_0^\epsilon$ was introduced to count the number of elements in $\mathbf{b} \in \mathbb{R}^n$, defined as $\| \mathbf{b} \|_0^\epsilon = card(\{i :| b_i |< \epsilon, i \in \forall n\})$, where b_i was the ith element of $\mathbf{b}$, and ϵ was set to 0.001. The cosparsity of the signal could be obtained by applying $\| \mathbf{b} \|_0^\epsilon$ to the product of the learned dictionary $\hat{\mathbf{\Omega}}$ and the original signal $\mathbf{Y}$. The recovery cosparsity is closer to the cosparsity l, the learned dictionary is better.

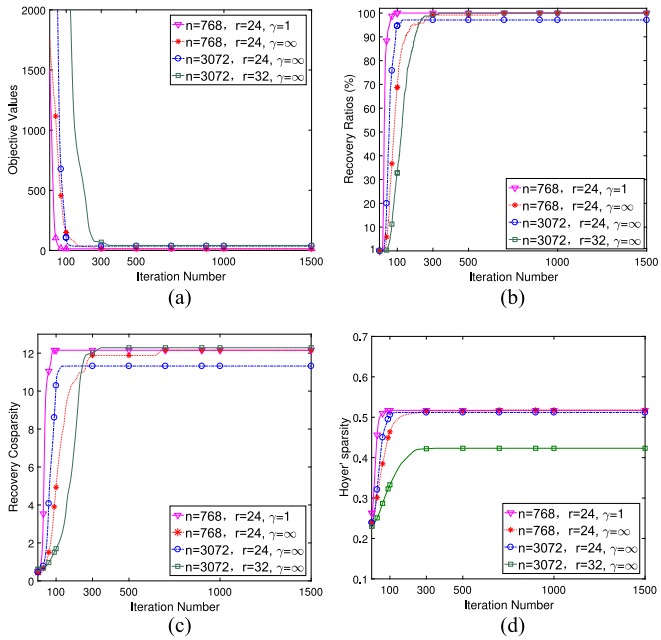

Fig. 1. (a) Objective values versus iteration number. (b) Average recovery ratios for the learned dictionaries versus iteration number. (c) Average recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus iteration number. (d) Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus iteration number.

In addition, Hoyer's measure [33] was used as the sparsity measure, which was absolute and had values in $[0, 1]$ for different sparsities monotonically. Thus, Hoyer's measure can be controlled explicitly. It is defined as follows:

$$\text{Sparsity}(\mathbf{X}) = \frac{\sqrt{n} - \|\mathbf{X}\|_1/\|\mathbf{X}\|_F}{\sqrt{n} - 1} \in [0, 1]. \tag{37}$$

A larger Hoyer's sparsity value means that the matrix $\mathbf{X}$ is sparser.

3) Convergence of the Proposed Algorithm: We tested the convergence behavior from independent trials with different n, r, and γ. The number n of signal samples was selected as 768 and 3072, and the number r of dictionary atoms was selected as 24 and 32. In this experiment, we considered not only increases in the numbers of atoms and signal samples but also different initial dictionaries; i.e., $\gamma = 1$ and $\gamma \to \infty$. In addition, the cosparsity was set to $l = 12$ in this example. To determine the convergence behavior of the proposed PADL-$\ell_{1/2}$ algorithm, we executed the algorithm several times to identify a reasonable number of iterations that would ensure the convergence of the algorithm. The objective values, the recovery ratios of the learned dictionary, the recovery cosparsities, and Hoyer's sparsity are shown in Fig. 1(a)–(d), respectively. For each point in the plots, the results were averaged over 30 independent trials. These results indicate that the objective function of PADL-$\ell_{1/2}$ decreases monotonically and converges to stable values within a reasonable number of iterations in different cases. Furthermore, the recovery ratios of the learned dictionary, the recovery

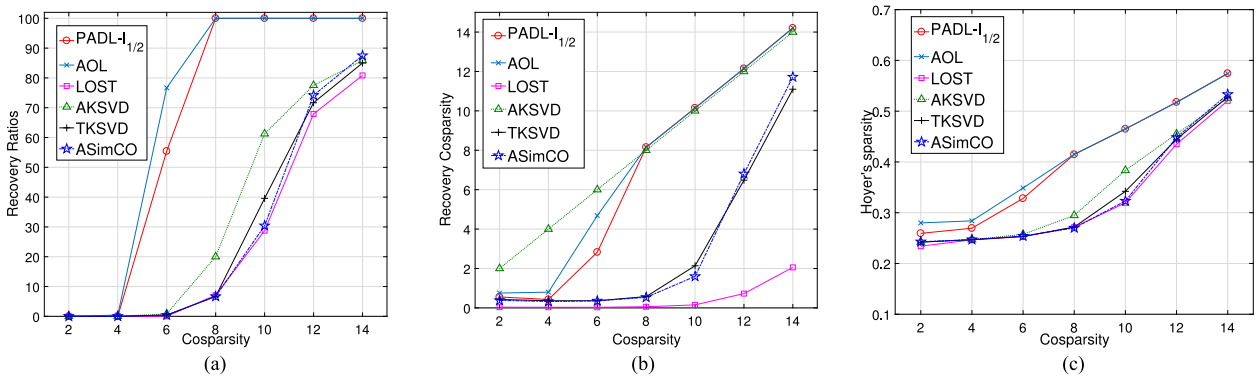

Fig. 2. $n = 768$, $r = 24$ and $\gamma = \infty$. (a) Average recovery ratios for the learned dictionaries versus cosparsity for the various algorithms. (b) Average recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms. (c) Average Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms.

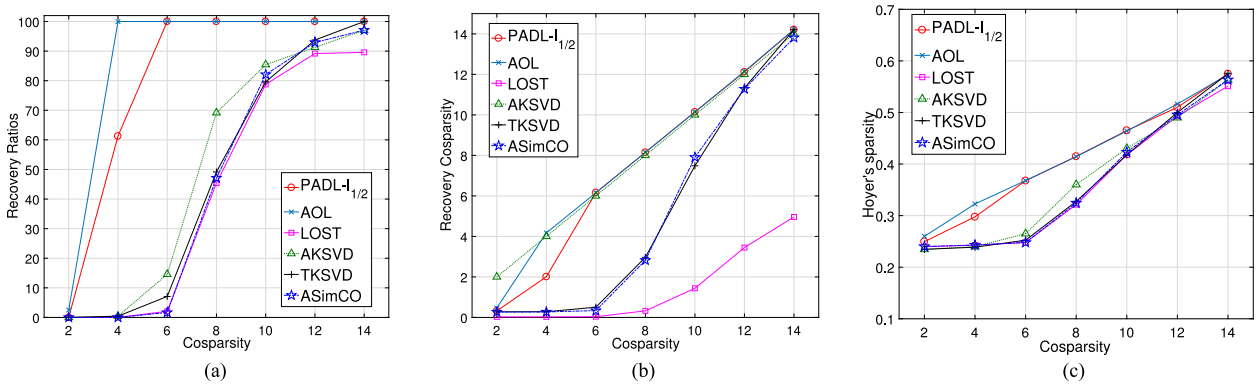

Fig. 3. $n = 3072$, $r = 24$ and $\gamma = \infty$. (a) Average recovery ratios for the learned dictionaries versus cosparsity for the various algorithms. (b) Average recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms. (c) Average Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms.

cosparsities of $\hat{\mathbf{\Omega}}\mathbf{Y}$, and Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ also converge to stable values. We also found that with more signal samples and fewer dictionary atoms, the proposed algorithm converged faster. In addition, the proposed algorithm with the initial dictionary $\gamma = 1$ converged faster than with the random dictionary $\gamma \to \infty$.

4) Exact Recovery of the Analysis Dictionary: We compared our proposed algorithm PADL-$\ell_{1/2}$ with five state-of-the-art algorithms: AKSVD [11], TKSVD [14], ASimCO [15], LOST [12], and AOL [13]. AKSVD, TKSVD LOST, and ASimCO use the ℓ_0 norm as their sparsity constraint. AOL uses the ℓ_1 norm as its sparsity constraint. All the algorithms were initialized with the same dictionary. The algorithms were tested with different values of the cosparsity, l, which was varied over the range $l = 2, ..., 14$. In addition, we investigated the role of n, r, and γ in the dictionary recovery process via some independent simulations. According to the settings of the step sizes and the regularization parameter of PADL-$\ell_{1/2}$ given in Section III-D, we chose $\mu^k = L_{\mathbf{x}}$ and $\nu^k = L_{\mathbf{\Omega}}$, and $\lambda_l^k = \sqrt{\frac{16}{27}}\mu_k T_{s+1}^{\frac{3}{2}}$. The settings for AKSVD, TKSVD, ASimCO, and AOL were the same as those used in the experiments with synthetic data, as described in [11], [13]–[15]. In LOST [12], two penalty parameters, the parameter λ of the full column rank penalty term and the parameter η of the correlation penalty term, required adjustments. λ was set to 50 and η was set to 20, so that a better dictionary for the synthetic data could be learned. The step size for the inner gradient conjugate method of LOST was 10^{-4}. We executed the algorithms as many times as possible and determined a reasonable number of iterations that could ensure that the objective function of each algorithm converged to stable values. In the experiments, we performed 500 iterations for AKSVD, 1000 iterations for TKSVD, 5000 iterations for ASimCO, LOST, and PADL-$\ell_{1/2}$, and 50,000 iterations for AOL.

In this experiment, the main performance indexes were the recovery ratios for the learned dictionaries, the recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$, and Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus the cosparsity l. We tested the performances of the algorithms not only with the increase of the numbers of atoms and the signal samples but also with different initial dictionaries. The results were averaged over 30 independent trials for each point in the plots.

Fig. 2 shows the average recovery ratios of the learned dictionaries, the average recovery cosparsity, and the average Hoyer's sparsity for the different algorithms when $n = 768$, $r = 24$, and $\gamma \to \infty$. Fig. 3 shows the average recovery ratios of the learned dictionaries, the average recovery cosparsity, and the average Hoyer's sparsity for the large-scale samples. The results show that better dictionaries can be learned with more signal samples. Fig. 4 shows the average recovery ratios of the learned dictionaries, the average recovery cosparsity, and the average Hoyer's sparsity for a large-scale dictionary. The results

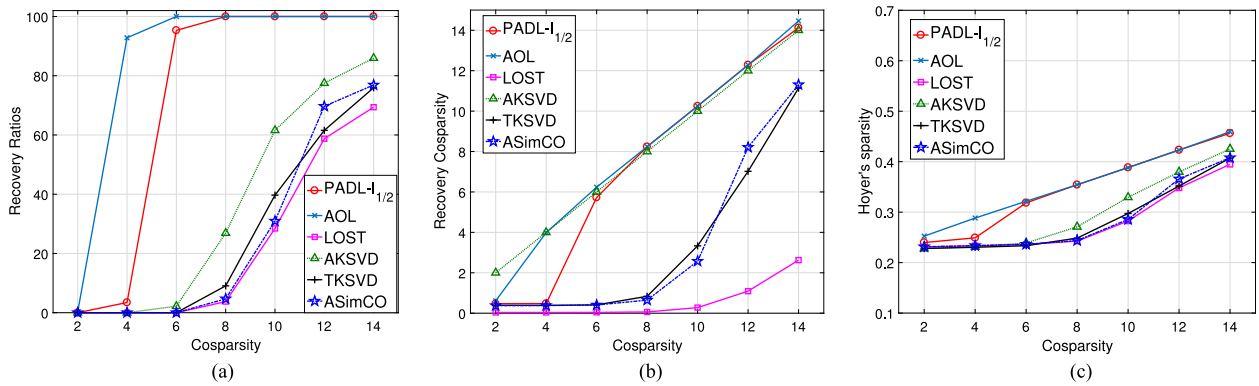

Fig. 4. $n = 3072$, $r = 32$ and $\gamma = \infty$. (a) Average recovery ratios for the learned dictionaries versus cosparsity for the various algorithms. (b) Average recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms. (c) Average Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus cosparsity for the various algorithms.

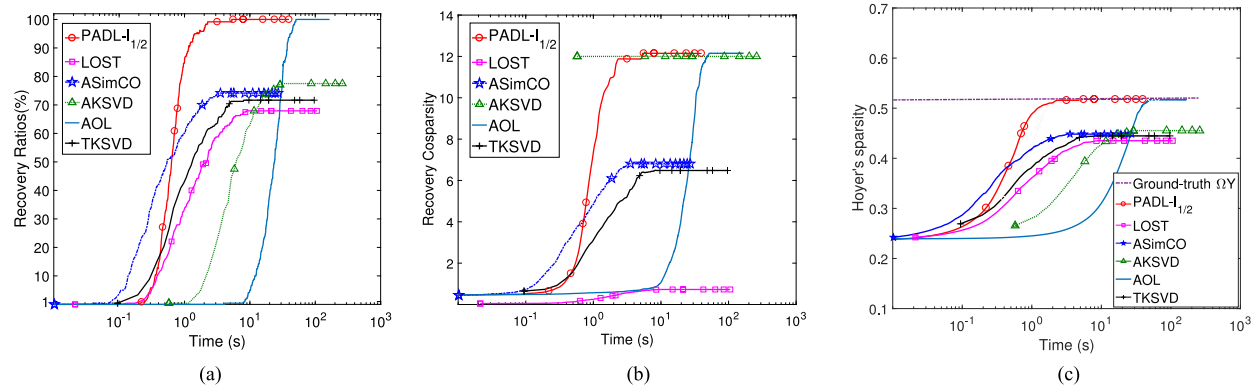

Fig. 5. $n = 768$, $r = 24$ and $\gamma = \infty$. (a) Average recovery ratios for the learned dictionaries versus running time for the various algorithms. (b) Average recovery cosparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus running time for the various algorithms. (c) Average Hoyer's sparsity of $\hat{\mathbf{\Omega}}\mathbf{Y}$ versus running time for the various algorithms.

demonstrate that the algorithms' performance becomes worse with increased numbers r of atoms.

We found that better results, including the recovery ratios of the learned dictionaries, the recovery cosparsity, and the Hoyer's sparsity, can be obtained for the algorithms with larger cosparsities, as shown in Figs. 2–4. In particular, our PADL-$\ell_{1/2}$ algorithm and AOL with the UNTF constraint have significantly better performance than the other algorithms, and moreover, PADL-$\ell_{1/2}$ and AOL are still effective even for low cosparsities, while the other algorithms degrade greatly. We also found that our PADL-$\ell_{1/2}$ algorithm has superior recovery ratios of the learned dictionaries, recovery cosparsity, and Hoyer's sparsity compared with AKSVD, TKSVD, ASimCO, and LOST. Note that AKSVD used the backward greedy algorithm so that it had the same recovery cosparsity as l throughout the iterations seen in Figs. 2(b), 3(b), and 4(b). We therefore used Hoyer's sparsity as another sparsity measure. It can be seen that Hoyer's sparsity for PADL-$\ell_{1/2}$ is larger than that for AKSVD. Note that our PADL-$\ell_{1/2}$ algorithm is slightly worse than AOL for low cosparsity. This might be because AOL is based on the analysis model where $\| \mathbf{\Omega}\mathbf{Y} \|_1$ is directly minimized, but its convergence is slower than that of PADL-$\ell_{1/2}$ based on the transform model, which can be verified by the following experimental results in Fig. 5.

Obviously, the algorithms converge differently in reaching stable values for the dictionary ratios, Hoyer's sparsity, and the recovery cosparsity. Keeping the above experimental settings, the case $l = 12$ was selected to visualize the convergence speeds for the algorithms. Note that different algorithms require different numbers of iterations to converge, and their running times per iteration are also different. Hence, to compare the algorithms fairly, the performances of the algorithms versus running time per iteration are shown in Fig. 5. We can see that the running times to reach stable values for PADL-$\ell_{1/2}$ are shorter than those for the state-of-the-art algorithms. In particular, the recovery ratios of the learned dictionaries, the recovery cosparsity, and Hoyer's sparsity are similar for PADL-$\ell_{1/2}$ and AOL, but the running time of convergence for PADL-$\ell_{1/2}$ is much shorter than that for AOL. In addition, we plotted the Hoyer's sparsity of the ground-truth $\mathbf{\Omega}\mathbf{Y}$ as comparison in Fig. 5(c). We can see that the Hoyer's sparsity of PADL-$\ell_{1/2}$ can reach to that of the ground-truth $\mathbf{\Omega}\mathbf{Y}$.

B. Sparse Approximation of Audio Data

In this experiment, we studied the sparse approximation for audio data. The audio data used in the experiments were BBC-radio music samples recorded at 16 kHz for 4.1 s. From that audio sample, 1024 pieces of signal were extracted as training samples, each of size 64×1.

To investigate the robustness of the $\ell_{1/2}$ norm compared with that of the ℓ_0 norm and the ℓ_1 norm, we used the ℓ_0 norm instead of the $\ell_{1/2}$ norm in PADL-$\ell_{1/2}$ and employed the proximal

TABLE I
COMPARISON OF HOYER'S SPARSITY AND NSE WITH DIFFERENT COSPARSITY l

	$l = 30$		$l = 40$		$l = 50$	
	Hoyer's sparsity	NSE	Hoyer's sparsity	NSE	Hoyer's sparsity	NSE
PADL-ℓ_0	0.2814	0.1346	0.3064	0.1927	0.3418	0.2845
PADL-ℓ_1	0.3433	0.8119	0.3861	0.8457	0.4152	0.8879
PADL-$\ell_{1/2}$	0.4639	0.0028	0.5484	0.0065	0.5847	0.0136

operator for the ℓ_0 norm—i.e., the hard-thresholding function [20]—to obtain the approximate analysis matrix $\mathbf{X}$. The procedure for learning the dictionary was retained from that in PADL-$\ell_{1/2}$. Therefore, we developed the algorithm and called it PADL-ℓ_0. As we know, TKSVD, ASimCO, and LOST use the ℓ_0 norm for sparsity and obtain the analysis matrix $\mathbf{X}$ by the hard-thresholding function which is an approximate approach. The procedure of obtaining $\mathbf{X}$ for PADL-ℓ_0 is the same with that for LOST, ASimCO and TKSVD. We compared the proposed algorithm with PADL-ℓ_0 instead of LOST, ASimCO and TKSVD. We also used the ℓ_1 norm instead of the $\ell_{1/2}$ norm in PADL-$\ell_{1/2}$ and adopted the proximal operator for the ℓ_1 norm—i.e., the soft-thresholding function [34]—to obtain the analysis matrix $\mathbf{X}$. The procedure for learning the dictionary was the same as that used with PADL-$\ell_{1/2}$. We called the algorithm PADL-ℓ_1. As we know, the soft-thresholding function obtains the solution in closed form, and the proximal operator for the $\ell_{1/2}$ norm also obtains the solution in closed form. The algorithm AOL use the ℓ_1 norm for sparsity, and has no the procedure for learning the analysis matrix $\mathbf{X}$. So we did not compare the proposed algorithm with AOL.

A pseudo-random admissible $\mathbf{\Omega}_0 \in \mathbb{R}^{128\times 64}$ was selected as the initial analysis dictionary. Using the training samples, we executed PADL-$\ell_{1/2}$ to learn the analysis dictionaries $\mathbf{\Omega}$ of size 128×64. With the learned dictionary, we executed PADL-ℓ_0, PADL-ℓ_1 and PADL-$\ell_{1/2}$ to obtain the sparse approximation $\mathbf{X}$. Various cosparsities l were selected.

To assess the algorithms fairly, we used the Hoyer's sparsity of $\mathbf{X}$ as the measure, as defined in (37). We also introduced the normalized sparsification error (NSE) to evaluate the quality of the sparse approximation $\mathbf{X}$, which was defined as:

$$\text{NSE} = \frac{\|\mathbf{\Omega Y} - \mathbf{X}\|_F^2}{\|\mathbf{\Omega Y}\|_F^2}. \tag{38}$$

The average results are summarized in Table I. Clearly, larger values of l result in larger values of Hoyer's sparsity and NSE. This demonstrates that a larger number l of zero elements in the analysis matrix $\mathbf{X}$ results in a larger Hoyer's sparsity and a larger error in the approximation representation. Importantly, PADL-$\ell_{1/2}$ can obtain higher values of Hoyer's sparsity and smaller values of NSE than PADL-ℓ_0 or PADL-ℓ_1 for the same level of l. The ℓ_0 norm can lead the strongest sparsity theoretically. However, the analysis sparse representation with the ℓ_0 norm is a combinatorial optimization problem and its complexity is of NP-hard. PADL-ℓ_0 employed the greed algorithm, so that their results are just some approximate solutions of the ℓ_0 norm. These approximate solutions cannot reach the sparsity of its theoretical expectation. In addition, The table demonstrates that the proximal operator for the $\ell_{1/2}$ norm can obtain more accurate results than the proximate operator for the ℓ_0 norm or that for the ℓ_1 norm. From the theoretical analysis [35], the proximal operator for the ℓ_0 norm is a greedy algorithm that only obtains approximate solutions. The proximal operator for the ℓ_1 norm can obtain the exact solution, but it is parallel to the identity, leading to high-valued estimates receiving large biases. However, the proximal operator for the $\ell_{1/2}$ norm not only obtains the exact solution but also converges to the identity asymptotically, ensuing that any high-valued estimates obtained are less biased and that the recovered results are more accurate compared with the estimates by the proximal operator for the ℓ_1 norm.

VI. CONCLUSION

We have proposed an efficient and fast algorithm for ADL, where the $\ell_{1/2}$ norm regularizer is used for strong sparsity. We employ the proximal alternating minimization scheme with global convergence property for solving the $\ell_{1/2}$ norm regularized optimization problems. The proposed algorithm is not only theoretically sound for non-convex problems arising from the $\ell_{1/2}$ norm, but also shown to have excellent performance in the experiments: (1) the proposed algorithm can recover the analysis dictionary and the cosparsity better, and can also achieve greater Hoyer's sparsity than the state-of-the-art algorithms. Furthermore, the proposed algorithm has smaller running times for convergence. (2) We have verified the robustness of the proximal operator for the $\ell_{1/2}$ norm compared with the proximal operators for the ℓ_0 and ℓ_1 norms in terms of the sparsity, the error of approximation representation.

In future work, we plan to apply our proposed algorithm to applications to evaluate further the performance of our method.

APPENDIX

In this appendix, we give a proof of Theorem 1.

Theorem 2: ([36]) Assume $\phi(z)$ is a proper and lower semi-continuous function with $\inf \phi > -\infty$, the sequence $\{z^k\}$ is bounded and converges to the critical points of the function $\phi(z)$, if the following conditions hold:
(V1) Sufficient decrease condition. There exists some positive constant a such that:

$$\phi(z^k) - \phi(z^{k+1}) \geq a \parallel z^{k+1} - z^k \parallel_2^2, \quad \forall k$$

(V2) Relative error condition. There exists some positive constant b such that:

$$\parallel \omega^{k+1} \parallel_2^2 \leq b \parallel z^{k+1} - z^k \parallel_2^2, \quad \forall k$$

(V3) Continuity condition. There exists a subsequence $\{z^{k_j}\}_{j\in\mathbb{N}}$ and $\bar{z}$ such that:

$$z^{k_j} \to \bar{z}, \quad \phi(z^{k_j}) \to \phi(\bar{z}), \qquad \text{as} \quad j \to \infty$$

(V4) $\phi(z)$ is a KL function [26]. $\phi(z)$ satisfies the KL property in its effective domain.

Next, we check that the sequence $\{\mathbf{Z}^k\} = \{\mathbf{X}^k, \mathbf{\Omega}^k\}$ generated by Algorithm 1 satisfies the conditions V1–V4. The objective function in our analysis dictionary learning $\phi(\mathbf{Z}) = F(\mathbf{X}) + Q(\mathbf{\Omega}, \mathbf{X}) + G(\mathbf{\Omega})$ is a proper and lower semi-continuous function, where $F(\mathbf{X}), Q(\mathbf{\Omega}, \mathbf{X})$ and $G(\mathbf{\Omega})$ have been defined in Section III-A.

Condition (V1): Proof: In our analysis dictionary learning problem (3), the analysis matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_l, ..., \mathbf{x}_n]$ is updated via the proximal alternating minimization algorithm, such that:

$$\mathbf{x}_l^{k+1} = \arg\min_{\mathbf{x}_l} \frac{\mu^k}{2} \left\| \mathbf{x}_l - \left(\mathbf{x}_l^k - \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) \right) \right\|_2^2 + F(\mathbf{x}_l) \quad \forall l. \tag{39}$$

hence, we have,

$$\frac{\mu^k}{2} \left\| \mathbf{x}_l^{k+1} - \left(\mathbf{x}_l^k - \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) \right) \right\|_F^2 + F(\mathbf{x}_l^{k+1}) \leq \frac{\mu^k}{2} \left\| \frac{1}{\mu^k} \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k)) \right\|_2^2 + F(\mathbf{x}_l^k), \tag{40}$$

then,

$$F(\mathbf{x}_l^k) \geq F(\mathbf{x}_l^{k+1}) + \frac{\mu^k}{2} \|\mathbf{x}_l^{k+1} - \mathbf{x}_l^k\|_2^2 + \langle \mathbf{x}_l^{k+1} - \mathbf{x}_l^k, \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) \rangle. \tag{41}$$

Furthermore, Q is the C^1 function and $\bigtriangledown Q$ is Lipschitz continuous on bounded subsets of $\mathbb{R}$. Then, we have

$$Q(\mathbf{x}_l^{k+1}, \mathbf{\Omega}^k) \leq Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) + \langle \mathbf{x}_l^{k+1} - \mathbf{x}_l^k, \bigtriangledown_{\mathbf{x}_l} Q(\mathbf{x}_l^k, \mathbf{\Omega}^k) \rangle + \frac{L_{\mathbf{x}}}{2} \|\mathbf{x}_l^{k+1} - \mathbf{x}_l^k\|_2^2 \quad \forall l. \tag{42}$$

where $L_{\mathbf{x}}$ is Lipschitz continuous of $\bigtriangledown_{\mathbf{x}} Q$. Then combine (41) and (42), we have

$$F(\mathbf{X}^{k+1}) + Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) \leq F(\mathbf{X}^k) + Q(\mathbf{X}^k, \mathbf{\Omega}^k) - \frac{\mu^k - L_{\mathbf{x}}}{2} \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2. \tag{43}$$

Again this inequality has been obtained owing to the separability of the Frobenius norm and the $\ell_{1/2}$ norm.

For the analysis dictionary $\mathbf{\Omega}$, we have,

$$Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^{k+1}) + G(\mathbf{\Omega}^{k+1}) \leq Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) + G(\mathbf{\Omega}^k) - \frac{\nu^k - L_{\mathbf{x}}}{2} \|\mathbf{\Omega}^{k+1} - \mathbf{\Omega}^k\|_F^2. \tag{44}$$

Summing up (43) and (44), we have

$$\phi(\mathbf{X}^k, \mathbf{\Omega}^k) - \phi(\mathbf{X}^{k+1}, \mathbf{\Omega}^{k+1}) \geq \frac{\mu^k - L_{\mathbf{x}}}{2} \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 + \frac{\nu^k - L_{\mathbf{x}}}{2} \|\mathbf{\Omega}^{k+1} - \mathbf{\Omega}^k\|_F^2. \tag{45}$$

The step sizes are chosen as $\mu^k \geq L_{\mathbf{x}_l} > 0$ and $\nu^k \geq L_{\mathbf{\Omega}} > 0$. Thus, condition (V1) for $\{\mathbf{Z}^k\} = \{\mathbf{X}^k, \mathbf{\Omega}^k\}$ is satisfied.

Condition (V2): Proof: According to (18), we have the iteration of $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_l, ..., \mathbf{x}_n]$ is obtained as

$$\mathbf{X}^{k+1} = \arg\min_{\mathbf{X}} F(\mathbf{X}) + \hat{Q}(\mathbf{X}) + \frac{\mu^k}{2} \|\mathbf{X} - \mathbf{X}^k\|_F^2, \tag{46}$$

where $\hat{Q}(\mathbf{X}^{k+1}) = Q(\mathbf{X}^k, \mathbf{\Omega}^k) + trace(\langle \mathbf{X}^{k+1} - \mathbf{X}^k, \bigtriangledown_{\mathbf{X}} Q(\mathbf{X}^k) \rangle)$. Note that (46) has been obtained owing to the separability of the Frobenius norm and the trace operator.

Therefore, we have

$$0 \in \partial F(\mathbf{X}^{k+1}) + \bigtriangledown_{\mathbf{X}} Q(\mathbf{X}^k, \mathbf{\Omega}^k) + \mu^k (\mathbf{X}^{k+1} - \mathbf{X}^k). \tag{47}$$

Then

$$\bigtriangledown_{\mathbf{X}} Q(\mathbf{Z}^{k+1}) - \bigtriangledown_{\mathbf{X}} Q(\mathbf{X}^k, \mathbf{\Omega}^k) - \mu^k (\mathbf{X}^{k+1} - \mathbf{X}^k) \in \partial_{\mathbf{X}} \phi(\mathbf{Z}^{k+1}). \tag{48}$$

Similarly, for $\mathbf{\Omega}$, according to (27), we have,

$$\bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{Z}^{k+1}) - \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) - \nu^k (\mathbf{\Omega}^{k+1} - \mathbf{\Omega}^k) \in \partial_{\mathbf{\Omega}} \phi(\mathbf{Z}^{k+1}). \tag{49}$$

Define $\omega^k := (\omega_{\mathbf{X}}^k, \omega_{\mathbf{\Omega}}^k)$. Then, we have

$$\omega_{\mathbf{X}}^{k+1} := \bigtriangledown_{\mathbf{X}} Q(\mathbf{Z}^{k+1}) - \bigtriangledown_{\mathbf{X}} Q(\mathbf{X}^k, \mathbf{\Omega}^k) - \mu^k (\mathbf{X}^{k+1} - \mathbf{X}^k) \in \partial_{\mathbf{X}} \phi(\mathbf{Z}^{k+1}), \tag{50}$$

$$\omega_{\mathbf{\Omega}}^{k+1} := \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{Z}^{k+1}) - \bigtriangledown_{\mathbf{\Omega}} Q(\mathbf{X}^{k+1}, \mathbf{\Omega}^k) - \nu^k (\mathbf{\Omega}^{k+1} - \mathbf{\Omega}^k) \in \partial_{\mathbf{\Omega}} \phi(\mathbf{Z}^{k+1}). \tag{51}$$

Then $\omega^k := (\omega_{\mathbf{X}}^k, \omega_{\mathbf{\Omega}}^k) \in \partial\phi(\mathbf{Z}^k)$. Additionally, by the boundedness of $\{\mathbf{Z}^k\} = \{\mathbf{X}^k, \mathbf{\Omega}^k\}$ and Lipschitz continuous of $\bigtriangledown Q$, there exists a constant $\rho > 0$ such that

$$\| \omega^{k+1} \|_F^2 \leq \rho \| \mathbf{Z}^{k+1} - \mathbf{Z}^k \|_F^2, \quad \forall k. \tag{52}$$

Thus, condition (V2) is satisfied.

Condition (V3): Proof: Given a positive integer j, from condition (V1), we have,

$$\phi(z^0) - \phi(z^j) \geq a \sum_{k=0}^{j} \| z^k - z^{k+1} \|_2^2, \tag{53}$$

Since $\phi(z^k)$ is decreasing and $\inf\phi > -\infty$, there exist some $\phi(\bar{z})$ such that $\phi(z^j) \to \phi(\bar{z})$ as $j \to +\infty$. When $j \to +\infty$, we have

$$\sum_{k=0}^{j \to +\infty} \| z^k - z^{k+1} \|_2^2 < \phi(z^0) - \phi(z^j) < +\infty, \tag{54}$$

this means $\lim \| z^k - z^{k+1} \|_2^2 = 0$.

From (46), we have

$$F(\mathbf{X}^{k+1}) + \hat{Q}(\mathbf{X}^{k+1}) + \frac{\mu^k}{2}\|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 \leq F(\mathbf{X}) + \hat{Q}(\mathbf{X}) + \frac{\mu^k}{2}\|\mathbf{X} - \mathbf{X}^k\|_F^2 \quad (55)$$

Let $k = k_j - 1$ and $\mathbf{X} = \bar{\mathbf{X}}$. By the Lipschitz continuity of ∇Q and condition (V1), we have $\limsup_{j\to+\infty} F(\mathbf{X}^{k_j}) \leq F(\bar{\mathbf{X}})$. Similarly, for $\mathbf{\Omega}$, we get $\limsup_{j\to+\infty} G(\mathbf{\Omega}^{k_j}) \leq G(\bar{\mathbf{\Omega}})$. Additionally, since F and G are lower semi-continuous, we have $\lim_{j\to+\infty} F(\mathbf{X}^{k_j}) = F(\bar{\mathbf{X}})$ and $\lim_{j\to+\infty} G(\mathbf{\Omega}^{k_j}) = G(\bar{\mathbf{\Omega}})$. Furthermore, Q is continuous, thus $\lim_{j\to+\infty} \phi(\mathbf{Z}^{k_j}) = \phi(\bar{\mathbf{Z}})$. Condition (V3) is satisfied.

There exists $\omega^{k_j} \in \partial\phi(\mathbf{Z}^{k_j})$. From (52) and (54), we have

$$\sum_{k=0}^{+\infty} \| \omega^{k_j} \|_F^2 \leq \rho\big(\phi(\mathbf{Z}^0) - \phi(\bar{\mathbf{Z}})\big) < +\infty, \quad (56)$$

which leads to $\lim_{j\to+\infty} \omega^{k_j} = 0$. $\{\phi(\mathbf{Z}^{k_j})\}$ is a decreasing sequence and $\phi(\mathbf{Z}^{k_j}) \geq 0$. As $\omega^{k_j} \to 0$, $\bar{Z}$ is a critical point of (3).

Theorem 3: ([36]) If f is semi-algebraic defined in [26], then it satisfies the KL property at any point in $\mathrm{dom}\, f$.

Condition (V4): Proof: Our function $\phi(\mathbf{Z}) = F(\mathbf{X}) + Q(\mathbf{\Omega}, \mathbf{X}) + G(\mathbf{\Omega})$ is a proper and lower semi-continuous function, which satisfies the KL property according to Theorem 3. Condition (V4) is satisfied.

REFERENCES

[1] M. Zhang and C. Desrosiers, "Image denoising based on sparse representation and gradient histogram," *IET Image Process.*, vol. 11, no. 1, pp. 54–63, 2017.

[2] G. Dong, G. Kuang, N. Wang, and W. Wang, "Classification via sparse representation of steerable wavelet frames on Grassmann manifold: Application to target recognition in SAR image," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2892–2904, Jun. 2017.

[3] R. Giryes, Y. Plan, and R. Vershynin, "On the effective measure of dimension in the analysis cosparse model," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5745–5753, Oct. 2015.

[4] Z. Yang, Y. Zhang, W. Yan, Y. Xiang, and S. Xie, "A fast non-smooth nonnegative matrix factorization for learning sparse representation," *IEEE Access*, vol. 4, pp. 5161–5168, 2016.

[5] Y. Deng, Q. Dai, and Z. Zhang, "An overview of computational sparse models and their applications in artificial intelligence," in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, vol. 427. New York, NY, USA: Springer, Jan. 2013, pp. 345–369.

[6] M. Elad, *Sparse and Redundant Representation*. Berlin, Germany: Springer, 2010.

[7] Z. Li, S. Ding, and Y. Li, "A fast algorithm for learning overcomplete dictionary for sparse representation based on proximal operators," *Neural Comput.*, vol. 27, no. 9, pp. 1951–1982, Jul. 2015.

[8] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "Cosparse analysis modeling—uniqueness and algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2011, pp. 5804–5807.

[9] X. Peng, Y. Tang, W. Du, and F. Qian, "Multimode process monitoring and fault detection: A sparse modeling and dictionary learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4866–4875, Jun. 2017.

[10] Z. Li, H. Huang, and S. Misra, "Compressed sensing via dictionary learning and approximate message passing for multimedia internet of things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 505–512, Apr. 2017.

[11] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.

[12] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 3088–3092.

[13] M. Yaghoobi, S. Nam, R. Gribonval, and M. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2341–2355, May 2013.

[14] E. Eksioglu and O. Bayir, "'K-SVD meets transform learning: Transform K-SVD," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 347–351, Mar. 2014.

[15] J. Dong, W. Wang, W. Dai, M. Plumbley, Z. Han, and J. Chambers, "Analysis SimCO algorithms for sparse analysis model based dictionary learning," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 417–431, Sep. 2015.

[16] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2138–2150, Jun. 2013.

[17] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *Int. J. Comput. Vis.*, vol. 114, no. 2–3, pp. 137–167, 2015.

[18] Y. Zhang, T. Yu, and W. Wang, "An analysis dictionary learning algorithm under a noisy data model with orthogonality constraint," *Sci. World J.*, vol. 2014, 2014, Art. no. 852978.

[19] Z. Li, S. Ding, T. Hayashi, and Y. Li, "Analysis dictionary learning using block coordinate descent framework with proximal operators," *Neurocomputing*, vol. 239, pp. 165–180, 2017.

[20] I. Daubechies, M. Defrise, and C. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[21] Z. Xu, H. Zhang, Y. Wang, and X. Chang, "$l_{1/2}$ regularizer," *Sci. China*, vol. 53, no. 6, pp. 1159–1169, 2010.

[22] L. Niu, R. Zhou, Y. Tian, Z. Qi, and P. Zhang, "Nonsmooth penalized clustering via ℓ_p regularized sparse regression," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1423–1433, Jun. 2017.

[23] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *J. Mach. Learn. Res.*, vol. 11, pp. 1081–1107, 2010.

[24] A. Rakotomamonjy, R. Flamary, and G. Gasso, "DC proximal Newton for nonconvex optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 636–647, Mar. 2016.

[25] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proc. Int. Conf. Mach. Learn.*, 2013, vol. 28, pp. 37–45.

[26] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1–2, pp. 459–494, Jul. 2014.

[27] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[28] Z. Li, T. Hayashi, S. Ding, Y. Li, and X. Li, "Constrained analysis dictionary learning with the $\ell_{1/2}$-norm regularizer," in *Proc. IEEE 13th Int. Conf. Signal Process.*, Nov. 2016, pp. 890–894.

[29] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms for image processing," in *Proc. IEEE Int. Conf. Image Process.*, 2012, pp. 681–684.

[30] J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *Comptes Rendus de l'Académie des Sciences (Paris), Série A Math.*, vol. 255, pp. 2897–2899, 1962.

[31] F. C. Xing, "Investigation on solutions of cubic equations with one unkonwn," *Natural Sci. Educ.*, vol. 12, no. 3, pp. 207–218, 2003.

[32] J. A. Tropp, I. S. Dhillon, R. W. Heath, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 188–209, Jan. 2005.

[33] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

[34] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[35] I. Bayram, P. Y. Chen, and I. W. Selesnick, "Fused lasso with a non-convex sparsity inducing penalty," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 4156–4160.

[36] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semialgebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Math. Program. Ser.*, vol. 137, no. 1–2, pp. 91–129, 2013.

Authors' photographs and biographies not available at the time of publication.