International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

Analysis Sparse Representation Based on Subset Pursuit and Weighted Split Bregman Iteration Algorithm

Regular Paper

Ye Zhang[1]*, Tenglong Yu[1] and Wenquan Zhang[1]

1 Department of Electronic Information Engineering, Nanchang University, Nanchnag, China
*Corresponding author(s) E-mail: zhangye@ncu.edu.cn

Received 23 January 2014; Accepted 09 September 2014

DOI: 10.5772/61543

Abstract

Recently, a sparse representation model - called an analysis sparse model - where the signal is multiplied by an analysis dictionary and the outcome is assumed to be sparse, has received increasing attention since it has potential and extensive applications in the area of signal processing. The performance of the analysis model significantly depends on an appropriately chosen dictionary. Most existing analysis dictionary learning algorithms are based on the assumption that the original signals are known or can be estimated from their noisy versions. Generally, however, the original signals are unknown or need to be estimated by using greedy-like algorithms with heavy computation. To solve the problems, we introduce a subset pursuit algorithm for analysis dictionary learning, where the observed signals are directly employed to learn the analysis dictionary. Next, a weighted split Bregman iteration algorithm is proposed to estimate original signals by the learned analysis dictionary. The experimental results demonstrate the competitive performance of the proposed algorithms compared with the state-of-art algorithms.

Keywords Sparse representation, synthesis model, analysis model, dictionary learning, image denoising

1. Introduction

Sparse representation has become a well-known topic in a wide range of fields, such as image processing [1], compressed sensing [2], sensor networks [3], robotics [4, 5, 6, 7, 8], and more. A popular model for sparse representation is the synthesis model. In this model, a signal $\mathbf{x} \in R^M$ is represented as $\mathbf{x} = \mathbf{D}\mathbf{a}$, where $\mathbf{D} \in R^{M \times N}$ is a possibly overcomplete dictionary $(N \geq M)$ and $\mathbf{a} \in R^N$ is the coefficient vector which is assumed to be sparse, i.e., $\|\mathbf{a}\|_0 = L \ll N$, where the ℓ_0 quasi-norm $\|\cdot\|_0$ counts the number of nonzero components in its argument. In the synthesis model, the signal $\mathbf{x}$ can be described as a linear combination of only a few columns (i.e., atoms) of $\mathbf{D}$. In the past decade, the synthesis model has been extensively studied [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. In robotics, face recognition is used to satisfy person identification tasks and sparse representation-based classification can deal with facial recognition very well. The basic idea is to combine the sub-dictionaries which are learned from various classes' images and then represent the query image using a small number of atoms indicating the class of the image [5]. Sparse representation-based classification can also be applied to action recognition for human-robot interaction [6, 7] and object classification for robot perception [8].

Recently, an alternative form of the sparse representation model called the 'analysis model' was proposed in [13, 14, 15, 16, 17, 18, 19, 20, 21]. In this model, an analysis dictionary (or analysis operator) $\mathbf{\Omega} \in R^{P \times M}$ $(P \geq M)$ is sought in order to transform the signal vector $\mathbf{x}$ to a high-dimensional space, i.e., $\mathbf{\Omega x}=\mathbf{z}$, where the analysis coefficient vector $\mathbf{z} \in R^{P}$ is the analysis representation of $\mathbf{x}$ and is assumed to be sparse, i.e., $\|\mathbf{z}\|_0 \ll P$, while the total number of the zeros of $\mathbf{z}$ characterizes co-sparsity [13]. In the model, the signal $\mathbf{x}$ is described by the zero elements of $\mathbf{z}$ - in other words, zero entries in $\mathbf{z}$ define a subspace that the signal $\mathbf{x}$ belongs to as opposed to the few non-zero entries of $\mathbf{a}$ in the synthesis model.

Similar to the synthesis model, the performance of the analysis model relies on the sparse representation of the signals with an appropriately chosen dictionary and a signal recovery method. Compared to the extensive studies of synthesis dictionary learning, however, the problem of analysis dictionary learning (ADL) has received much less attention, with only a few algorithms proposed recently [15, 16, 17, 18, 20, 21]. Based on the fact that a row of $\mathbf{\Omega}$ is orthogonal to a subset $\mathbf{X}_j$ of signals $\mathbf{X}$, where $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_K] \in R^{M \times K}$, a sequential minimal eigenvalue algorithm has been proposed for ADL [15]. Once $\mathbf{X}_j$ is found, the corresponding j-th row of the dictionary can be updated with the eigenvector associated with the smallest eigenvalue of the Gram matrix of $\mathbf{X}_j$. However, the computational cost of this ADL method increases considerably if P becomes larger. In fact, the original signals $\mathbf{X}$ cannot be accurately observed. In[17, 16], a projected sub-gradient algorithm is proposed for analysis operator learning and an augmented Lagrangian (AL) algorithm is utilized to recover the signals $\mathbf{X}$. In this algorithm, however, a uniformly normalized tight frame is also employed as a constraint on the dictionary to avoid the trivial solution. This constraint limits the possible $\mathbf{\Omega}$ to be learned. In [18], the analysis K-SVD (AK-SVD) algorithm was proposed for ADL using the observed signals $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_K] \in R^{M \times K}$ measured in the presence of additive noise as

$$\mathbf{Y} = \mathbf{X} + \mathbf{V} \tag{1}$$

where $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_K] \in R^{M \times K}$ is a zero-mean white-Gaussian additive noise matrix. The optimization scheme in the AK-SVD algorithm is based on a two-phase alternating iterative approach. In the first phase, the optimal backward greedy (OBG) algorithm is applied to estimate $\mathbf{X}$ while keeping $\mathbf{\Omega}$ fixed. In the second phase, $\mathbf{\Omega}$ is updated by using the estimation of $\mathbf{X}$. The AK-SVD algorithm is effective but the phase of pre-estimate $\mathbf{X}$ leads to a computationally slow process. The learning over-complete sparsifying transform (LOST) algorithm was proposed for ADL by directly using noisy signal $\mathbf{Y}$ [20]. It is noted in the LOST algorithm that the null dictionary matrix can also lead to a trivial solution to the optimization problem. To eliminate such a trivial solution, the full-rank constraint on $\mathbf{\Omega}$ is employed in [20], but the ill-conditioned $\mathbf{\Omega}^T\mathbf{\Omega}$ may degrade the performance of the LOST algorithm.

In this paper, we introduce a subset pursuit algorithm to learn the analysis dictionary by directly employing the observed data to compute the approximate analysis sparse representation of the original signals [22]. The algorithm eliminates the need for estimating the original signals as otherwise required in the algorithms mentioned above. According to the absolute values of the analysis sparse representation, the observed data can be assigned into multiple subsets which are then used to update the analysis dictionary. To improve the performance of the signal recovery with the learned dictionary, a weighted split Bregman iterative (WSBI) algorithm is proposed to reconstruct the original signals in the analysis model, where the ℓ_1 minimization has been replaced by the weighted ℓ_1 minimization to promote the sparsity of the analysis representation of $\mathbf{x}$. The simulations' results demonstrate the competitive performance of the proposed algorithm compared with those of the state-of-art algorithms, such as AK-SVD [18], the noise-aware analysis operator learning algorithm (NAAOLA) [16, 17], and the LOST [20] and OBG [18] algorithms.

The paper is organized as follows. In Section 2, the analysis model and subset pursuit algorithm for ADL are described. In Section 3, the WSBI algorithm is proposed to reconstruct the original signals. Some experimental results are shown in Section 4, before concluding the paper in Section 5.

2. Subset pursuit algorithm for analysis dictionary learning

In this section, we present a subset pursuit algorithm for ADL by directly using the observed data without having to pre-estimate $\mathbf{X}$ (as done in [18]). More specifically, we exploit the analysis representation of $\mathbf{Y}$ to obtain the subset $\mathbf{Y}_j$ rather than using the estimation of $\mathbf{X}$ to determine $\mathbf{Y}_j$.

2.1 The analysis model

The analysis model can be described as follows: for a signal $\mathbf{x} \in R^{M}$ and a fixed redundant analysis dictionary $\mathbf{\Omega} \in R^{P \times M}$ $(P > M)$, the co-sparsity l of the analysis model is

$$l = P - \|\mathbf{\Omega x}\|_0 \tag{2}$$

The quantity l denotes the number of zeros in the vector $\mathbf{\Omega}x$, which implies that l rows in $\mathbf{\Omega}$ are orthogonal to the signal $\mathbf{x}$, and these rows define the co-support Λ, i.e., $\mathbf{\Omega}_\Lambda x=0$, where $\mathbf{\Omega}_\Lambda$ is a sub-matrix of $\mathbf{\Omega}$ that contains the rows from $\mathbf{\Omega}$ indexed by Λ. In this case, the signal $\mathbf{x}$ is said to be l-co-sparse and characterized by its co-support Λ. It is clear that the dimension of the subspace that signal $\mathbf{x}$ resides in is $r = M - l$. Generally, we can assume that $\mathbf{X}$ has the same co-rank $M - r$ related to the dictionary $\mathbf{\Omega}$.

2.2 Subset pursuit algorithm

Suppose we measure a signal of the form

$$\mathbf{y}_i = \mathbf{x}_i + \mathbf{v}_i \quad i = 1,2,\ldots,K \tag{3}$$

where $\mathbf{v}_i$ is a zero-mean white-Gaussian additive noise vector with a bounded ℓ_2 norm - say $\|\mathbf{v}_i\|_2 \leq \sigma$ - where σ denotes the noise level. According to Eq.(2), the task of ADL can be formed as

$$\min\|\boldsymbol{\Omega}\mathbf{x}_i\|_0 \tag{4}$$

This problem is NP-complete [2, 23]. Just like in the synthesis case, one might replace the ℓ_0 quasi-norm with the ℓ_1 norm

$$\min\|\boldsymbol{\Omega}\mathbf{x}_i\|_1 \tag{5}$$

where $\|\cdot\|_1$ is the ℓ_1 norm that sums the absolute values of a vector. In general, if the noise $\mathbf{v}_i$ is stationary and bounded, the analysis model Eq.(5) has an approximate analysis model

$$\min\|\boldsymbol{\Omega}\mathbf{y}_i\|_1 \tag{6}$$

Proof: Using Eq.(6), the analysis model is therefore written as

$$\mathbf{z}_i = \boldsymbol{\Omega}\mathbf{y}_i \tag{7}$$

where $\mathbf{z}_i=[z_{1i}z_{2i}\cdots z_{Pi}]^T \in R^{P\times 1}$ is the analysis representation of $\mathbf{y}_i$. Considering a single row ω_j^T in the analysis dictionary $\boldsymbol{\Omega}$, Eq.(7) can be rewritten as

$$z_{ij} = \boldsymbol{\omega}_j^T\mathbf{y}_i \tag{8}$$

Accordingly, the absolute value of z_{ji} is

$$|z_{ji}| = |\boldsymbol{\omega}_j^T\mathbf{y}_i| = |\boldsymbol{\omega}_j^T\mathbf{x}_i + \boldsymbol{\omega}_j^T\mathbf{v}_i| \tag{9}$$

We know that $|\boldsymbol{\omega}_j^T\mathbf{x}_i + \boldsymbol{\omega}_j^T\mathbf{v}_i| \leq |\boldsymbol{\omega}_j^T\mathbf{x}_i| + |\boldsymbol{\omega}_j^T\mathbf{v}_i|$. In general, the absolute values of z_{ji} have a small value when $\boldsymbol{\omega}_j$ is orthogonal to the signal $\mathbf{x}_i$, i.e., $\boldsymbol{\omega}_j^T\mathbf{x}_i$=0. Thus, the analysis sparse representation of $\mathbf{y}_i$ can be used to determine whether $\boldsymbol{\omega}_j^T$ is orthogonal to $\mathbf{x}_i$. ■

Therefore, we can compute the analysis sparse representation of the observed data $\mathbf{y}_i$ to find whether $\boldsymbol{\omega}_j^T$ is orthogonal with the signal $\mathbf{x}_i$, rather than using $\hat{\mathbf{x}}_i$ which otherwise has to be estimated from $\mathbf{y}_i$. Because the co-rank is assumed to be $M-r$ - which implies that $M-r$ rows in $\boldsymbol{\Omega}$ may be orthogonal to the data $\mathbf{y}_i$ - we can regard $M-r$ smallest values in $|\boldsymbol{\Omega}\mathbf{y}_i|$ as zeros. The $\Lambda_i := \{j \mid |\boldsymbol{\omega}_j^T\mathbf{y}_i| \approx 0\}$, which is the co-support of $\mathbf{y}_i$, can be obtained by the locations of the zero entries in $\boldsymbol{\Omega}\mathbf{y}_i$. As such, $\mathbf{y}_i$ can be assigned into the sub-set $\mathbf{Y}_j, \forall j \in \Lambda_i$. After $\mathbf{Y}_j$ is found, ω_j is updated as follows [18]:

$$\hat{\boldsymbol{\omega}}_j = \arg\min_{\boldsymbol{\omega}_j}\|\boldsymbol{\omega}_j^T\mathbf{Y}_j\|_2^2 \text{ s.t. } \|\boldsymbol{\omega}_j\|_2 = 1 \tag{10}$$

For the optimization problem, $\boldsymbol{\omega}_j$ can be updated using the eigenvector associated with the smallest eigenvalue of $\mathbf{Y}_j\mathbf{Y}_j^T$. This algorithm is described in the **Algorithm 1** table.

Algorithm 1: SP-ADL
Input: Observed data $\mathbf{Y} \in R^{M\times K}$, the initial dictionary $\boldsymbol{\Omega}_0 \in R^{P\times M}$, the co-rank $M-r$ and the number of iterations T
Output: Dictionary $\boldsymbol{\Omega}$
Initialization: Set $\boldsymbol{\Omega} := \boldsymbol{\Omega}_0$, Let $\mathbf{Y}'$ be the column-normalized version of $\mathbf{Y}$, where $\mathbf{Y}' = [\mathbf{y}'_1,\ldots,\mathbf{y}'_K] \in R^{M\times K}$
For $t = 1,\ldots,T$ **do**
  For $i = 1,\ldots,K$ **do**

- Compute $\mathbf{z}_i = \boldsymbol{\Omega}\mathbf{y}'_i$, select $M-r$ numbers of $|z_{ji}|$ which have the smallest values and find the cosupport Λ_i
- Assign corresponding $\mathbf{y}_i$ into $\mathbf{Y}_j, \forall j \in \Lambda_i$

  End for
  For $j = 1\ldots P$ **do**
    Update $\boldsymbol{\omega}_j$ with Eq.(10)
  End for
End for

3. The WSBI algorithm for recovering original signals based on the analysis model

In the analysis model, the estimation of $\mathbf{x}$ can be obtained from its noise version $\mathbf{y}$ by solving the optimization problem

$$\arg\min_{\mathbf{x}}\|\boldsymbol{\Omega}\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y}-\mathbf{x}\|_2^2 \leq \varepsilon \tag{11}$$

where ε is an estimated upper bound on the noise power $\|\mathbf{v}\|_2^2$. The OBG algorithm [18] and some pursuit methods, such as analysis iterative hard thresholding (AIHT), analysis hard thresholding pursuit (AHTP), analysis subspace pursuit (ASP)and analysis compressive sampling matching pursuit (ACoSaMP), have been proposed to solve the optimization problem [24]. Due to the use of the greedy-like methods, the estimation of $\mathbf{x}$ is computationally slow and also becomes unreliable with any increase of noise in $\mathbf{y}$. In [17], the AL method is applied to estimate the original signal $\mathbf{x}$. Indeed, the AL method is identical to the split Bregman iteration (SBI) algorithm [25]; however, the simulation results demonstrate that the performance of the AL algorithm is limited [17]. To solve the signal recovery problem, a WSBI algorithm is proposed whereby the weighted ℓ_1 minimization would lead to a more accurate solution than that obtained by the ℓ_1 minimization [26]. To begin with in the next section, we introduce the SBI algorithm, while the WSBI is proposed subsequently.

3.1 Split Bregman iterative algorithm

Recovering $\mathbf{x}$ in the analysis model is based on the optimization problem Eq.(11), and the constrained optimization problem can be transformed into an unconstrained optimization problem with the Lagrange multiplier μ. This leads to the following optimization problem:

$$\arg\min_{\mathbf{x}}\left(\|\mathbf{\Omega x}\|_1+\frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2\right) \tag{12}$$

Setting $\mathbf{z}=\mathbf{\Omega x}$, Eq.(12) can be written as

$$\arg\min_{\mathbf{z},\mathbf{x}}\left(\|\mathbf{z}\|_1+\frac{\mu}{2}\|\mathbf{y}-\mathbf{\Omega x}\|_2^2\right)\text{subject to } \mathbf{z}=\mathbf{\Omega x} \tag{13}$$

For simplicity, we set $E(\mathbf{z},\mathbf{x})=\|\mathbf{z}\|_1+\frac{\mu}{2}\|\mathbf{y}-\mathbf{\Omega x}\|_2^2$, and as such Eq.(13) can be transformed into the following problem:

$$\arg\min_{\mathbf{z},\mathbf{x}} E(\mathbf{z},\mathbf{x})\text{ subject to } \mathbf{z}=\mathbf{\Omega x} \tag{14}$$

The Lagrange method can also be used to transform the constrained optimization problem (14) into an unconstrained one as follows,

$$\arg\min_{\mathbf{z},\mathbf{x}}\left(E(\mathbf{z},\mathbf{x})+\frac{\eta}{2}\|\mathbf{z}-\mathbf{\Omega x}\|_2^2\right) \tag{15}$$

where η is Lagrange multiplier. In order to apply the SBI algorithm, we introduce the Bregman distance which is defined as [27]

$$\begin{aligned}D_E^{\mathbf{q}^t}(\mathbf{z},\mathbf{x},\mathbf{z}^t,\mathbf{x}^t)&=E(\mathbf{z},\mathbf{x})-E(\mathbf{z}^t,\mathbf{x}^t)-\left\langle \mathbf{q}_{\mathbf{z}}^t,\mathbf{z}-\mathbf{z}^t\right\rangle\\&-\left\langle \mathbf{q}_{\mathbf{x}}^t,\mathbf{x}-\mathbf{x}^t\right\rangle\end{aligned} \tag{16}$$

where $(\mathbf{q}_{\mathbf{z}}^t,\mathbf{q}_{\mathbf{x}}^t)=\partial E(\mathbf{z}^t,\mathbf{x}^t)$, $\langle\cdot\rangle$ stands for the inner product operator and t is the number of iterations. The unconstrained optimization problem (15) is equivalent to the following problem:

$$\arg\min_{\mathbf{z},\mathbf{x}}\left(D_E^{\mathbf{q}^t}\left(\mathbf{z},\mathbf{x},\mathbf{z}^t,\mathbf{x}^t\right)+\frac{\eta}{2}\|\mathbf{z}-\mathbf{\Omega x}\|_2^2\right) \tag{17}$$

The variates in Eq.(17) are separable and the SBI algorithm can be used to solve the problem (17) as [25]

$$\mathbf{x}^{t+1}=\arg\min_{\mathbf{x}}\left(\frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2+\frac{\eta}{2}\|\mathbf{z}^t-\mathbf{\Omega x}-\mathbf{b}^t\|_2^2\right) \tag{18}$$

$$\mathbf{z}^{t+1}=\arg\min_{\mathbf{z}}\left(\|\mathbf{z}\|_1+\frac{\eta}{2}\|\mathbf{z}-\mathbf{\Omega x}^{t+1}-b^t\|_2^2\right) \tag{19}$$

$$\mathbf{b}^{t+1}=\mathbf{b}^t+\mathbf{\Omega x}^{t+1}-\mathbf{z}^{t+1} \tag{20}$$

Generally, the initial values of $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{b}$ are equal to zero vectors. The convergence of the SBI algorithm has been proved in [25], and it stops when the stopping criterion, i.e., $\frac{\|\mathbf{x}^{t+1}-\mathbf{x}^t\|_2}{\|\mathbf{x}^t\|_2}\le\rho$, is met, where $\rho>0$ is an arbitrary small constant. The SBI algorithm is summarized in the **Algorithm 2** table.

Algorithm 2: SBI

Input: Observed signals $\mathbf{y}\in R^M$, the analysis dictionary $\mathbf{\Omega}\in R^{P\times M}$, μ,η
Output: Estimated original signal $\hat{\mathbf{x}}\in R^M$
Initialization: Set $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{b}$ equal to zero vectors
Repeat
$\mathbf{x}^{t+1}=\arg\min_{\mathbf{x}}\left(\frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2+\frac{\eta}{2}\|\mathbf{z}^t-\mathbf{\Omega x}-\mathbf{b}^t\|_2^2\right)$
$\mathbf{z}^{t+1}=\arg\min_{\mathbf{z}}\left(\|\mathbf{z}\|_1+\frac{\eta}{2}\|\mathbf{z}-\mathbf{\Omega x}^{t+1}-\mathbf{b}^t\|_2^2\right)$
$\mathbf{b}^{t+1}=\mathbf{b}^t+\mathbf{\Omega x}^{t+1}-\mathbf{z}^{t+1}$
until $\frac{\|\mathbf{x}^{t+1}-\mathbf{x}^t\|_2}{\|\mathbf{x}^t\|_2}\le\rho$
End

3.2 Weighted split Bregman iterative algorithm

To improve the performance of the SBI algorithm, the ℓ_1 minimization can be replaced by the weighted ℓ_1 minimization based on the synthesis model [28]. In this subsection, the WSBI algorithm has been introduced for analysis sparse representation. The WSBI algorithm is based on the weighted ℓ_1 minimization, which can promote sparsity and improve the performance of sparse representation. The optimization problem (12) can be transformed into a new form as

$$\arg\min_{\mathbf{x}}\left(\|\mathbf{\Omega x}\|_{\mathbf{w},1}+\frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2\right) \tag{21}$$

In Eq.(21), $\|\cdot\|_{\mathbf{w},1}$ is the weighted ℓ_1 norm, i.e., $\|\mathbf{z}\|_{\mathbf{w},1}=\sum_{i=1}^{P}w_i|z_i|$, where w_i and z_i are the i -th elements of the weighting coefficient vector $\mathbf{w}$ and signal $\mathbf{z}$, respectively. The iteration algorithm for the problem (21) is similar to the problem (12) as

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \left(\frac{\mu}{2} \|\mathbf{y}-\mathbf{x}\|_2^2 + \frac{\eta}{2} \|\mathbf{z}^t - \mathbf{\Omega x} - \mathbf{b}^t\|_2^2 \right) \tag{22}$$

$$\mathbf{z}^{t+1} = \arg\min_{\mathbf{z}} \left(\|\mathbf{z}\|_{\mathbf{w},1} + \frac{\eta}{2} \|\mathbf{z} - \mathbf{\Omega x}^{t+1} - \mathbf{b}^t\|_2^2 \right) \tag{23}$$

$$\mathbf{b}^{t+1} = \mathbf{b}^t + \mathbf{\Omega x}^{t+1} - \mathbf{z}^{t+1} \tag{24}$$

The initial values of $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{b}$ are equal to zero vectors and the weighting coefficients are equal to one. Next, the weighting coefficients are updated by $w_i = 2\delta / (\delta + |z_i|)$, where δ is a standard deviation of $\mathbf{z}$ [29]. In order to solve the least squares problem (22), the first-order derivative of the cost function in Eq.(22) is set to zero and then $\mathbf{x}$ is updated by

$$\mathbf{x}^{t+1} = \left(\mu\mathbf{I} + \eta\mathbf{\Omega}^T\mathbf{\Omega}\right)^{-1} \left(\mu\mathbf{y} + \eta\mathbf{\Omega}^T\left(\mathbf{z}^t - \mathbf{b}^t\right)\right) \tag{25}$$

In each iteration, $\mathbf{z}$ can be obtained by thresholding $(\mathbf{\Omega x}^{t+1} + \mathbf{b}^t)$ as

$$z_i^{t+1} = \begin{cases} (\mathbf{\Omega x}^{t+1} + b^t)_i - \frac{w_i}{\eta}, & (\mathbf{\Omega x}^{t+1} + b^t)_i > \frac{w_i}{\eta}; \\ 0, & -\frac{w_i}{\eta} < (\mathbf{\Omega x}^{t+1} + b^t)_i < \frac{w_i}{\eta}; \\ (\mathbf{\Omega x}^{t+1} + b^t)_i + \frac{w_i}{\eta}, & (\mathbf{\Omega x}^{t+1} + b^t)_i < -\frac{w_i}{\eta}. \end{cases} \tag{26}$$

The stopping criterion of the WSBI algorithm is the same as that of the SBI algorithm. The proposed WSBI algorithm is summarized in the **Algorithm 3** table.

Algorithm 3: WSBI

Input: Observed signals $\mathbf{y} \in R^M$, the analysis dictionary $\mathbf{\Omega} \in R^{P\times M}$, μ, η and $k_{\max}$

Output: Estimated original signal $\hat{\mathbf{x}} \in R^M$

Initialization: Set $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{b}$ equal to zero vectors, the weighting coefficients equal to one

While $k < k_{\max}$ **do**

Repeat

$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \left(\frac{\mu}{2} \|\mathbf{y}-\mathbf{x}\|_2^2 + \frac{\eta}{2} \|\mathbf{z}^t - \mathbf{\Omega x} - \mathbf{b}^t\|_2^2 \right)$

$\mathbf{z}^{t+1} = \arg\min_{\mathbf{z}} \left(\|\mathbf{z}\|_{\mathbf{w},1} + \frac{\eta}{2} \|\mathbf{z} - \mathbf{\Omega x}^{t+1} - \mathbf{b}^t\|_2^2 \right)$

$\mathbf{b}^{t+1} = \mathbf{b}^t + \mathbf{\Omega x}^{t+1} - \mathbf{z}^{t+1}$

until $\frac{\|\mathbf{x}^{t+1}-\mathbf{x}^t\|_2}{\|\mathbf{x}^t\|_2} \le \rho$

End

$w_i = \frac{2\delta}{\delta+|z_i|}$

$k = k + 1$

End

4. Computer simulations

In this section, two categories of the experimental results are presented. In the first part, we present experiments on synthetic data to show the ability of the proposed SP-ADL algorithm to recover a dictionary that has been used to produce the set of training data, and we then consider the real images and a piecewise-constant (PWC) image de-noising problem. In the second part, the image de-noising performances of the proposed WSBI algorithm are also tested on real images and a PWC image based on the learned dictionary. In these experiments, $\mathbf{\Omega}_0 \in R^{P\times M}$ is the initial dictionary in which each row is orthogonal to a random set of $M-1$ training data and is also normalized [18].

4.1 Experiments for the SP-ADL algorithm

4.1.1 Recovering the ground-truth analysis dictionary

In the experiments, we use the same experiment protocol as in [18]. $\mathbf{\Omega} \in R^{50\times 25}$ is generated with random Gaussian entries and the training dataset consists of $K = 50000$ signals for each co-rank $l = 21$ with both a noise-free setup and a noisy setup ($\sigma = 0.04, SNR = 25$ dB). If $\min_i (1 - |\hat{\boldsymbol{\omega}}_i^T \boldsymbol{\omega}_j|) < 0.01$, a row $\boldsymbol{\omega}_i^T$ in the true dictionary $\mathbf{\Omega}$ is regarded as recovered, where $\hat{\boldsymbol{\omega}}_i^T$ is an atom of the trained dictionary. The co-rank is considered to be known. Using the SP-ADL algorithm to recover the true dictionary $\mathbf{\Omega}$, the results are presented in Fig. 1 and Fig. 2. It can be observed from Fig. 1(a) and Fig. 2(a) that the SP-ADL algorithm is convergent after 300 iterations, and 90% of the rows in the true dictionary $\mathbf{\Omega}$ are reconstructed for the noise-free setup and 78% for the noisy one. Compared with the state-of-art algorithms, i.e., the AK-SVD, NAAOLA and LOST algorithms, their parameters are set the same as those in the original works, and it can be seen from Fig. 1(b) -(d) that these algorithms are convergent after 100, 200 and 300 iterations for the AK-SVD, NAAOLA and LOST algorithms, respectively[1]. Therefore, these algorithms have different iteration numbers. After 100, 200 and 300 iterations for the AK-SVD, NAAOLA and LOST algorithms, it is observed from Fig. 2(b)-(d) that 92%, 40% and 66% of the rows in the true dictionary $\mathbf{\Omega}$ are recovered for the noise-free setup and 84%, 2% and 64% for the noisy one, respectively. The experimental results demonstrate that the recovery percentage of the SP-ADL algorithm is relatively higher than those of the NAAOLA and LOST algorithms but is slightly lower than that of the AK-SVD algorithm. However, the running time in each iteration of the AK-SVD algorithm is significantly higher than that of the SP-ADL algorithm. The total runtime of our algorithm for 300 iterations is about 3,297 and 3,276 seconds for the noise-free and the noisy cases respectively. In contrast, the AK-SVD algorithm takes

1 The terms used to measure the convergence of these algorithms are slightly different in these original works.

about 30,030 and 30,380 seconds respectively for only 100 iterations (Computer OS: Windows XP, CPU: Pentium(R) Dual-Core T4300 @ 2.10 GHz; RAM 1.96 GB). The main reason is that $\mathbf{X}$ does not need to be estimated for our algorithm in each iteration, as opposed to the AK-SVD algorithm.

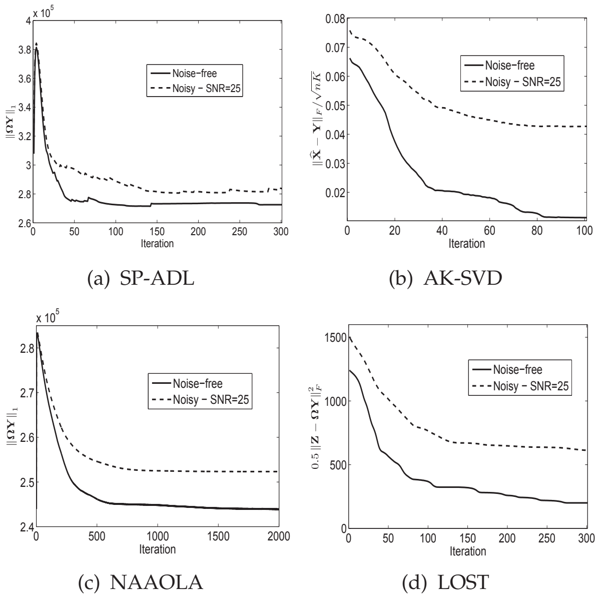


Figure 1. The convergent curves of the ADL algorithms

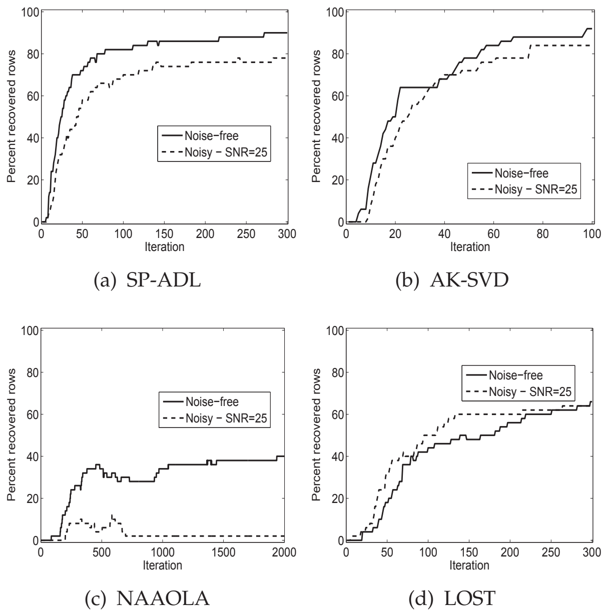


Figure 2. The recovery percentage curves of the ADL algorithms

4.1.2 Image de-noising

Using a test set consisting of four images commonly used in de-noising (Lena, House, Peppers and PWC), we compare the image de-noising performance of the SP-ADL algorithm with those of the AK-SVD, NAAOLA and LOST algorithms. In this experiment, the de-noising performance is evaluated by the peak signal-to-noise ratio (PSNR), defined as $\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2 KM}{\sum_{i=1}^{K}\sum_{j=1}^{M}(\hat{x}_{ij} - x_{ij})^2}\right)$ (dB), where x_{ij} and $\hat{x}_{ij}$ are the ij th pixel values in the noisy and de-noising images respectively.

In this experiment, a training set of 20,000 image patches each of 7×7 pixels, obtained from four images contaminated by noise with different noise levels σ, varying from 5 to 20, is employed for ADL by using the SP-ADL, AK-SVD, NAAOLA and LOST algorithms, respectively. The dictionary of size 63×49 is learned from the training data and the co-rank is assumed as $l=7$. The examples of the learned analysis dictionaries are shown in Fig.3. Based on the learned dictionaries, the OBG algorithm is utilized to recover the images from their noise versions. The parameters of the OBG algorithm are set the same as those in [18]. In the NAAOLA algorithm, the parameters are set the same as the original works except that λ=3(σ=5), λ=1(σ=10,15) and λ=0.5(σ=20), and $\alpha=10^{-11}$, $\lambda=\eta=10^5$ and q=20 in the LOST algorithm. The results, averaged over five trials, are presented in Table 1. It can be observed from Table 1 that the SP-ADL algorithm outperforms the NAAOLA and LOST algorithms. The de-noising performance of the SP-ADL algorithm is better than the AK-SVD algorithm if the noise level is increased.

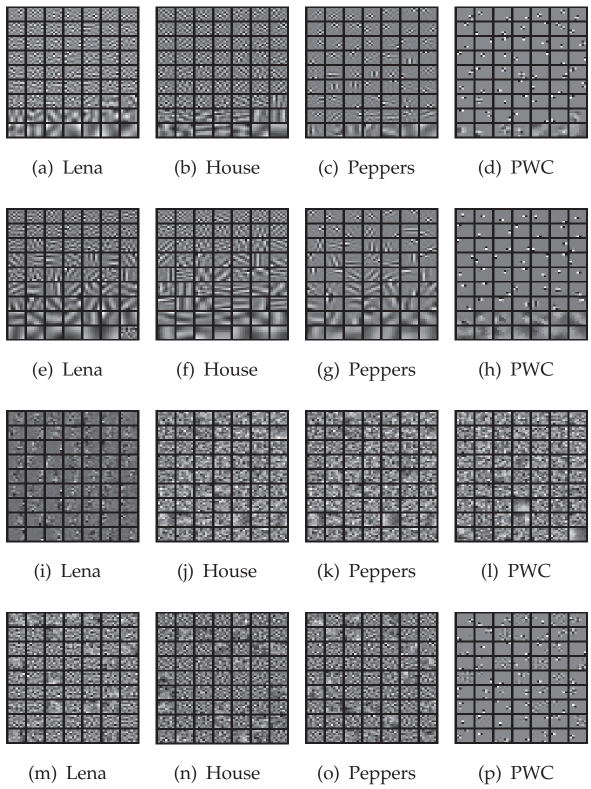
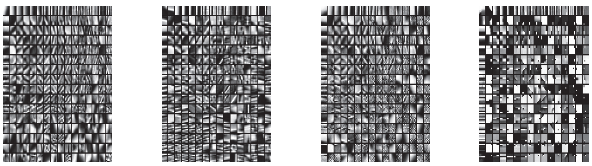


Figure 3. The learned analysis dictionaries of size 63×49 using the SP-ADL, AK-SVD, NAAOLA and LOST algorithms on the four images with a noise level δ=5. The SP-ADL algorithms' results are shown on the top row followed by the AK-SVD, NAAOLA and LOST algorithms.

4.1.3 Comparison with the synthesis model

In this experiment, the K-SVD algorithm - which is a state-of-art synthesis dictionary learning algorithm - is applied for image de-noising. The parameters of the K-SVD algorithm are set the same as those in [30]. The synthesis dictionary $\mathbf{D} \in R^{49\times256}$ is learned from the same training data used in ADL. The examples of the learned synthesis dictionaries are shown in Fig. 4. Employing the learned dictionary $\mathbf{D}$, the original images are reconstructed using the OMP algorithm [30]. The results, averaged over five trials, are presented in Table 1, from which we can see that the image de-noising performance of the K-SVD algorithm is better than that of the analysis methods. When the noise level $\sigma=5$, however, the image de-noising performance of the SP-ADL and AK-SVD algorithms outperforms the K-SVD algorithm for the images of Peppers and the PWC. The results show that the analysis method provides a better modelling platform than the synthesis method.



(a) Lena (b) House (c) Peppers (d) PWC

Figure 4. The learned synthesis dictionaries of size 49×256 using the K-SVD algorithm on the four images with the noise level δ=5

σ	Noisy	Method	Lena	House	Peppers	PWC
		SP-ADL	38.39	38.87	37.80	44.45
		AK-SVD	38.45	39.17	37.97	45.42
5	34.15	NAAOLA	37.36	37.04	35.93	37.28
		LOST	38.16	38.50	37.47	42.42
		K-SVD	38.63	39.60	37.75	43.63
		SP-ADL	35.07	35.19	33.87	37.61
		AK-SVD	34.84	35.34	33.83	38.40
10	28.13	NAAOLA	32.73	32.61	31.12	31.61
		LOST	34.75	34.77	33.61	36.43
		K-SVD	35.53	36.08	34.83	39.26
		SP-ADL	33.20	33.16	31.65	33.53
		AK-SVD	32.57	32.98	31.28	32.23
15	24.61	NAAOLA	30.87	30.82	29.22	29.48
		LOST	32.88	32.82	31.38	32.58
		K-SVD	33.66	34.38	33.31	35.59
		SP-ADL	31.90	31.88	30.05	30.53
		AK-SVD	31.42	31.53	29.77	29.59
20	22.11	NAAOLA	29.65	29.22	27.40	27.24
		LOST	31.64	31.47	29.84	29.93
		K-SVD	32.29	33.14	32.20	32.87

Table 1. Image de-noising results (PSNR dB)

4.2 Experiments for the WSBI algorithm

We employ the SP-ADL algorithm to learn the analysis dictionary and then the learned dictionaries are utilized to recover the signals by using the WSBI, SBI and OBG algorithms. In the WSBI algorithm, the parameters are set as $\eta=0.1$, $\rho=10^{-4}$, $k_{\max}=2$ and $\mu=0.3(\sigma=5)$,. $0.1(\sigma=10)$, $0.05(\sigma=15)$,$0.03(\sigma=20)$ The parameters of the SBI algorithm are the same as those in the WSBI algorithm. In the OBG algorithm, the parameters are set as those originally suggested in [18]. The results, averaged over five trials, are presented in Table 2 and Table 3.

σ	Noisy	Method	Lena	House	Peppers	PWC
		WSBI	38.12	38.78	37.35	42.28
5	34.15	SBI	37.66	38.28	36.95	40.38
		OBG	38.39	38.87	37.80	44.45
		WSBI	34.89	35.24	33.60	36.49
10	28.13	SBI	34.46	34.83	33.25	35.04
		OBG	35.08	35.18	33.87	37.61
		WSBI	33.13	33.50	31.54	32.75
15	24.61	SBI	32.94	33.15	31.26	31.46
		OBG	33.20	33.16	31.65	33.53
		WSBI	32.00	32.18	30.13	29.57
20	22.11	SBI	31.82	31.90	29.91	28.75
		OBG	31.90	31.88	30.05	30.53

Table 2. Image de-noising results (PSNR dB)

σ	Noisy	Method	Lena	House	Peppers	PWC
		WSBI	688	126	157	11
5	34.15	SBI	48	9	10	5
		OBG	3374	727	847	616
		WSBI	491	93	119	18
10	28.13	SBI	45	8	9	6
		OBG	2500	536	712	583
		WSBI	354	69	107	94
15	24.61	SBI	40	7	10	7
		OBG	2022	442	603	526
		WSBI	325	61	93	96
20	22.11	SBI	37	7	10	9
		OBG	1680	378	526	525

Table 3. Image de-noising results (Time sec.)

From Table 2, it can be seen that the image de-noising performance of the WSBI algorithm is better than that of the SBI algorithm. The simulation results demonstrate that the image de-noising performance can be improved using weighted ℓ_1 minimization. The WSBI algorithm is equivalent to the SBI algorithm when the weights are one, so the first outer iteration of the WSBI algorithm performs the same as the SBI algorithm. As such, WSBI algorithm is inevitably slower than the standard SBI algorithm, as shown in Table 3. From Table 2, the experimental results show that the image de-noising performance of the WSBI algorithm is similar to that of the OBG algorithm. However, the running time of the WSBI algorithm is far less than that of the OBG algorithm, as shown in Table 3.

5. Conclusion

For the analysis sparse representation model, we present a new algorithm for ADL and for recovering the original signal. The analysis dictionary is learned directly from the observed noisy data using the SP-ADL algorithm, and then the learned dictionary is utilized to recover the original signal using the WSBI algorithm where the ℓ_1 minimization is replaced by the weighted ℓ_1 minimization. As with the synthesis sparse representation, the analysis sparse representation model can also be applied in wide fields, especially face, gesture and action recognition in robotics.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61162014, 61210306074) and the Natural Science Foundation of Jiangxi (Grant No. 20122BAB201025, 20151BAB207005).

7. References

[1] M Aharon, M Elad, and A Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] Donoho and L David. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[3] J Liang, M Zhang, X Zeng, and G Yu. Distributed dictionary learning for sparse representation in sensor networks. *IEEE Transactions on Image Processing*, 23(6):2528 – 2541, 2014.

[4] B Argall, S Chernova, M Veloso, and B Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[5] D Zhong, P Zhu, J Han, and S Li. An improved robust sparse coding for face recognition with disguise. *International Journal of Advanced Robotic Systems*, 9:126, 2012.

[6] A Boyali and N Hashimoto. Block-sparse representation classification based gesture recognition approach for a robotic wheelchair. In *Intelligent Vehicles Symposium Proceedings, IEEE*, pages 2209–2215, September 2014.

[7] R Sean, G Ilaria, M Giorgio, and O Francesca. Keep it simple and sparse: real-time action recognition. *The Journal of Machine Learning Research*, 14(1):2617–2640, 2013.

[8] Z Kira. Transfer of sparse coding representations and object classifiers across heterogeneous robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1133–1138, June 2014.

[9] K Engan, S Aase, and J Husoy. Method of optimal directions for frame design. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2443–2446, 1999.

[10] K Skretting and K Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, 2010.

[11] W Dai, T Xu, and W Wang. Simultaneous codeword optimization (simco) for dictionary update and learning. *IEEE Transactions on Signal Processing*, 60(12):6340–6353, 2011.

[12] M Yaghoobi, T Blumensath, and M Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transactions on Signal Processing*, 57(6):2178–2191, 2009.

[13] M Elad, P Milanfar, and R Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947–968, 2007.

[14] S Nam, M Davies, M Elad, and R Gribonval. Cosparse analysis modeling - uniqueness and algorithms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5804–5807, 2011.

[15] B Ophir, M Elad, N Bertin, and M Plumbley. Sequential minimal eigenvalues: An approach to analysis dictionary learning. In *European Signal Processing Conference*, pages 1465–1469, 2011.

[16] M Yaghoobi, S Nam, R Gribonval, and M Davies. Noise aware analysis operator learning for approximately cosparse signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5409–5412, 2012.

[17] M Yaghoobi, S Nam, R Gribonval, and M Davies. Constrained overcomplete analysis operator learning for cosparse signal modelling. *IEEE Transactions on Signal Processing*, 61(9):2341–2355, 2013.

[18] R Rubinstein, T Peleg, and M Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3):661–677, 2013.

[19] S Ravishankar and Y Bresler. Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, 61(5):1072–1086, 2013.

[20] S Ravishankar and Y Bresler. Learning overcomplete sparsifying transforms for signal processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3088–3092, 2013.

[21] Y Zhang, H Wang, W Wang, and S Sanei. K-plane clustering algorithm for analysis dictionary learning. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–4, 2013.

[22] Y Zhang, H Wang, T Yu, and W Wang. Subset pursuit for analysis dictionary learning. In *European Signal Processing Conference*, pages 1–5, 2013.

[23] J Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

[24] R Giryes, S Nam, M Elad, R Gribonval, and M Davies. Greedy-like algorithms for the cosparse analysis model. *Linear Algebra and its Applications*, 441:22–60, 2013.

[25] J Cai, S Osher, and Z Shen. Split bregman methods and frame based image restoration. *Multiscale modeling & simulation*, 8(2):337–369, 2009.

[26] E Candes, M Wakin, and S Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.

[27] W Yin, S Osher, D Goldfarb, and J Darbon. Bregman iterative algorithms for l_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

[28] Y Zhao and D Li. Reweighted l_1-minimization for sparse solutions to underdetermined linear systems. *SIAM Journal on Optimization*, 22(3):1065–1088, 2012.

[29] S Arberet, P Vandergheynst, R Carrillo, J Thiran, and Y Wiaux. Sparse reverberant audio source separation via reweighted analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7): 1391–1402, 2013.

[30] M Elad and M Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IE EE Transactions on Image Processing*, 15(12):3736–3745, 2006.