AN ANALYSIS DICTIONARY LEARNING ALGORITHM BASED ON RECURSIVE LEAST SQUARES

Ye Zhang[1], *Haolong Wang*[1], *Wenwu Wang*[2]

[1]Department of Electronic and Information Engineering, Nanchang University, Nanchang, China
[2]Department of Electronic Engineering, University of Surrey, Guildford, United Kingdom
Emails: zhangye@ncu.edu.cn, hlwangncu@163.com, w.wang@surrey.ac.uk

ABSTRACT

We consider the dictionary learning problem in sparse representations based on an analysis model with noisy observations. A typical limitation associated with several existing analysis dictionary learning (ADL) algorithms, such as Analysis K-SVD, is their slow convergence due to the procedure used to pre-estimate the source signal from the noisy measurements when updating the dictionary atoms in each iteration. In this paper, we propose a new ADL algorithm where the recursive least squares (RLS) algorithm is used to estimate the dictionary directly from the noisy measurements. To improve the convergence properties of the proposed algorithm, the initial dictionary is estimated from a small training set by using the K-plane clustering algorithm. The proposed algorithm, as shown by experiments, offers advantages over the Analysis K-SVD, in both the runtime and atom recovery rate.

***Index Terms*—** Recursive least squares; analysis dictionary learning; sparse representation

1. INTRODUCTION

In the past decade spare representations based on the synthesis model have been studied extensively. Consider a signal $\mathbf{x} \in R^M$, the sparse representations of $\mathbf{x}$ over a dictionary $\mathbf{D}$ can be described as $\mathbf{x} = \mathbf{Da}$, where $\mathbf{D} \in R^{M\times N}$ is a possibly overcomplete dictionary ($N \geq M$), and $\mathbf{a} \in R^N$, containing the coding coefficients, is assumed to be sparse, i.e. $\|\mathbf{a}\|_0 = n \ll N$, where the ℓ_0 quasi-norm $\|\cdot\|_0$ counts the number of nonzero components in its argument. In other words, the signals are represented as a linear combination of a small number of signal components (i.e. atoms) chosen from the dictionary. A great deal of effort has been dedicated to the problem of learning the dictionary $\mathbf{D}$ from signal examples [1–9] based on the synthesis model.

There is a dual viewpoint to sparse representations based on e.g. the analysis model, where an analysis dictionary or operator $\mathbf{\Omega} \in R^{P\times M}$ ($P \geq M$) is sought to transform $\mathbf{x} \in R^M$ to a high dimensional space, i.e. $\mathbf{\Omega x} = \mathbf{z}$. The coefficient vector $\mathbf{z} \in R^P$ is called the analysis representation of $\mathbf{x}$ and assumed to be sparse. In this model, the rows of $\mathbf{\Omega}$ that are associated with zero entries in $\mathbf{z}$ define a subspace that the signal $\mathbf{x}$ is orthogonal to, as opposed to the few non-zero entries of $\mathbf{a}$ in the synthesis model. In general, $\mathbf{\Omega}$ can be learned from the observed signals $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2\ ...\ \mathbf{y}_K] \in R^{M\times K}$ measured in the presence of additive noise, i.e. $\mathbf{Y} = \mathbf{X} + \mathbf{V}$, where $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_K] \in R^{M\times K}$ contains the original signals, $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_K] \in R^{M\times K}$ is noise and K is the number of signals. Compared to the synthesis counterpart, there are only a few algorithms proposed recently for analysis dictionary learning (ADL) [10–13].

In [10], the Analysis K-SVD algorithm was proposed for dictionary learning. By keeping $\mathbf{\Omega}$ fixed, the optimal backward greedy (OBG) algorithm has been employed to estimate the sub-matrix of $\mathbf{\Omega}$ whose rows are orthogonal to $\hat{\mathbf{X}}$, which is the estimation of $\mathbf{X}$ from $\mathbf{Y}$. A data set, i.e., the sub-matrix of $\mathbf{Y}$, can be obtained, with its columns corresponding to that of $\hat{\mathbf{X}}$. The smallest singular values of the sub-matrix are then used to update $\mathbf{\Omega}$. The algorithm is effective, however, the original signal $\mathbf{X}$ should be estimated by using the greedy algorithm with heavy computation. In [11], a subset pursuit algorithm has been proposed for the ADL. The algorithm is based on the two-phase approach. In the first phase, the algorithm exploits the analysis representation of $\mathbf{Y}$ to obtain the subset $\mathbf{Y}_j$, rather than using $\hat{\mathbf{X}}_j$ to determine $\mathbf{Y}_j$, where $\mathbf{Y}_j$ is the subset of $\mathbf{Y}$ that corresponds to the *j*-th row of $\mathbf{\Omega}$, which is orthogonal to $\mathbf{Y}_j$. In the second phase, the *j*-th row of the analysis dictionary can be updated by the eigenvector associated with the smallest eigenvalue of the Gram matrix of $\mathbf{Y}_j$. In [12], a sequential minimal eigenvalue algorithm for the ADL was proposed by considering the orthogonality between the rows of $\mathbf{\Omega}$ and a sub-set of training signals $\mathbf{X}$. Once the sub-set is found, the corresponding row of the dictionary can be updated with the eigenvector associated with the smallest eigenvalue of the autocorrelation matrix of these signals. However, as the number of the rows in $\mathbf{\Omega}$ increases, so does the computational cost of the method. In [13], a projected subgradient algorithm was proposed for analysis operator learning, where an ℓ_1-norm penalty function is applied to $\mathbf{\Omega x}$, and a uniformly normalized tight frame is employed as a constraint on the dictionary to avoid the trivial solutions. However, the possible $\mathbf{\Omega}$ to be learned is restricted due to a rather arbitrary constraint enforced for the learning problem.

Most of the ADL algorithms mentioned above are based on the assumption that $\mathbf{X}$ is known or can be accurately estimated from its noisy version $\mathbf{Y}$. However, to estimate $\mathbf{X}$ by using the greedy algorithm is a computationally slow process and also becomes unreliable when the noise level is high. In this paper, we propose a new RLS algorithm for the ADL. In the algorithm, the dictionary can be learned directly from the noisy measurements without pre-estimating $\mathbf{X}$ when updating the atoms of the dictionary in each iteration. To further improve the convergence of the RLS-ADL algorithm, we use the K-plane clustering algorithm [14] to estimate an initial dictionary with a small number of training vectors, which is then utilised for the initialisation of the RLS-ADL algorithm. The proposed algorithm has advantages over the Analysis K-SVD algorithm, in particular, in terms of the runtime efficiency and the recovery rate of the dictionary atoms, as confirmed in our simulations.

The remainder of the paper is organized as follows. The proposed algorithm is detailed in Section 2. Simulations are provided in Section 3 to show its performance, and a brief conclusion is given in Section 4.

2. THE PROPOSED RLS-ADL ALGORITHM

2.1. The Analysis Model with Noisy Observations

In the analysis model, a dictionary (or operator) $\mathbf{\Omega}$ is sought such that $\mathbf{\Omega x} = \mathbf{z}$, and $\mathbf{z}$ is sparse, with its cosparsity (resembling the term of "sparsity" in the synthesis model) l denoted as

$$l = P - \|\mathbf{\Omega x}\|_0 \tag{1}$$

It implies that l rows in $\mathbf{\Omega}$ are orthogonal to $\mathbf{x}$. These rows define a cosupport Λ, i.e. $\mathbf{\Omega}_\Lambda \mathbf{x} = 0$, where $\mathbf{\Omega}_\Lambda \in R^{l\times M}$ 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 characterized by the zeros of the vector $\mathbf{\Omega x}$, and these zeros define the subspace that the signal belongs to. The task can be formed as

$$\min \|\mathbf{\Omega x}\|_0 \tag{2}$$

To solve this problem is NP-complete [15]. Hence the ℓ_0 quasi-norm is often replaced by the ℓ_1 norm

$$\min \|\mathbf{\Omega x}\|_1 \tag{3}$$

where $\|\cdot\|_1$ is the ℓ_1 norm that sums the absolute values of a vector. In the analysis model, if the true cosupport Λ is known, the signal $\mathbf{x}$ can be recovered from its noisy version $\mathbf{y} = \mathbf{x} + \mathbf{v}$ by [10]

$$\hat{\mathbf{x}} = \left(\mathbf{I} - \mathbf{\Omega}_\Lambda^\dagger \mathbf{\Omega}_\Lambda\right)\mathbf{y} \tag{4}$$

Different from [10], we aim to adapt $\mathbf{\Omega}$ directly from $\mathbf{y}$ without having to pre-estimate $\mathbf{x}$ using (4) when updating the atoms in each iteration. We propose an RLS-ADL algorithm for the update of the dictionary, as discussed next.

2.2. The Recursive Least Squares Algorithm

Similar to most dictionary learning algorithms, the proposed algorithm is a two-step process alternating between sparse coding and dictionary update.

2.2.1. *Estimating the coefficents* $\mathbf{z}$

In the proposed RLS-ADL algorithm, we use $\mathbf{\Omega y}_k$ to approximate $\mathbf{\Omega x}_k$, $k = 1, \cdots, K$. As shown in our previous work [14], such an approximation enables a computationally efficient optimization process while retaining similar results to those based on $\mathbf{\Omega x}_k$. Therefore, we set $\mathbf{z}_k = \mathbf{\Omega y}_k$, and form the following optimization criterion

$$\arg\min_{\mathbf{z}_k} \|\mathbf{z}_k - \mathbf{\Omega y}_k\|_2^2 + \gamma\|\mathbf{z}_k\|_1 \tag{5}$$

where $\mathbf{z}_k = [z_{1k} \;\; z_{2k} \;\; \cdots \;\; z_{Pk}]^T \in R^{P\times 1}$ and $\gamma > 0$ is a regularization parameter for enforcing sparsity. Given $\mathbf{\Omega}$, the first-order optimality condition of $\mathbf{z}_k$ implies that

$$\mathbf{z}_k = \mathbf{\Omega y}_k - \gamma \mathrm{sign}(\mathbf{z}_k) \tag{6}$$

It is well known that the above solution for $\mathbf{z}_k$ is called soft thresholding. Generally, since the cosparsity is assumed to be l, which implies the l rows in $\mathbf{\Omega}$ may be orthogonal to $\mathbf{y}_k$, we can regard l smallest absolute values in $\mathbf{z}_k$ as zeros.

2.2.2. *Updating the dictionary* $\mathbf{\Omega}$

Given $\mathbf{Z}$, the cost function used to estimate $\mathbf{\Omega}$ can be written as follows:

$$\arg\min_{\mathbf{\Omega}}(\|\mathbf{Z} - \mathbf{\Omega Y}\|_F^2) \tag{7}$$

The above cost function can be optimized in an online manner based on the RLS-ADL algorithm. To this end, we denote the collection of the first k training vectors as $\mathbf{Y}_k = [\mathbf{y}_1 \;\; \mathbf{y}_2 \;\; \cdots \;\; \mathbf{y}_k] \in R^{M\times k}$, and their corresponding coefficient vectors as $\mathbf{Z}_k = [\mathbf{z}_1 \;\; \mathbf{z}_2 \;\; \cdots \;\; \mathbf{z}_k] \in R^{P\times k}$. With the RLS-ADL updating rule, the dictionary at the k-th iteration can be obtained as

$$\mathbf{\Omega}_k = \mathbf{Z}_k \mathbf{Y}_k^T (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1} = \mathbf{B}_k \mathbf{C}_k \tag{8}$$

where $\mathbf{B}_k = \mathbf{Z}_k \mathbf{Y}_k^T$ and $\mathbf{C}_k = (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1}$.

To obtain the dictionary at the $(i + 1)$-th iteration, i.e. $\mathbf{\Omega}_{k+1}$, we can split the cost function (7) into two parts,

$$\mathbf{\Omega}_{k+1} = \arg\min(\lambda \|\mathbf{Z}_k - \mathbf{\Omega}_k \mathbf{Y}_k\|_F^2 + \|\mathbf{z}_{k+1} - \mathbf{\Omega}_k \mathbf{y}_{k+1}\|_2^2) \tag{9}$$

where $\mathbf{y}_{k+1}$ is a new training vector and $\mathbf{z}_{k+1}$ is the corresponding coefficient vector. Here, a forgetting factor λ is introduced to take into account the impact of the previous training samples on the current estimate, thereby improving the convergence performance of the RLS-ADL algorithm. Based on the RLS-ADL algorithm, we can define the recursion:

$$\mathbf{Z}_{k+1} = [\sqrt{\lambda}\mathbf{Z}_k, \mathbf{z}_{k+1}] \tag{10}$$

$$\mathbf{Y}_{k+1} = [\sqrt{\lambda}\mathbf{Y}_k, \mathbf{y}_{k+1}] \tag{11}$$

We can then update $\mathbf{B}_{k+1}$ and $\mathbf{C}_{k+1}$ as:

$$\begin{aligned}\mathbf{B}_{k+1} = \mathbf{Z}_{k+1}\mathbf{Y}_{k+1}^T = [\sqrt{\lambda}\mathbf{Z}_k, \mathbf{z}_{k+1}][\sqrt{\lambda}\mathbf{Y}_k, \mathbf{y}_{k+1}]^T \\ = \lambda\mathbf{Z}_k\mathbf{Y}_k^T + \mathbf{z}_{k+1}\mathbf{y}_{k+1}^T = \lambda\mathbf{B}_k + \mathbf{z}_{k+1}\mathbf{y}_{k+1}^T\end{aligned} \tag{12}$$

$$\begin{aligned}\mathbf{C}_{k+1}^{-1} = \mathbf{Y}_{k+1}\mathbf{Y}_{k+1}^T = [\sqrt{\lambda}\mathbf{Y}_k, \mathbf{y}_{k+1}][\sqrt{\lambda}\mathbf{Y}_k, \mathbf{y}_{k+1}]^T \\ = \lambda\mathbf{Y}_k\mathbf{Y}_k^T + \mathbf{y}_{k+1}\mathbf{y}_{k+1}^T = \lambda\mathbf{C}_k^{-1} + \mathbf{y}_{k+1}\mathbf{y}_{k+1}^T\end{aligned} \tag{13}$$

and then

$$\mathbf{C}_{k+1} = \lambda^{-1}\mathbf{C}_k - \frac{\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1}\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k}{\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1} + 1} \tag{14}$$

The dictionary at the $(k+1)$-th iteration is therefore updated as

$$\begin{aligned}\mathbf{\Omega}_{k+1} = \mathbf{B}_{k+1}\mathbf{C}_{k+1} = \\ (\lambda\mathbf{B}_k + \mathbf{z}_{k+1}\mathbf{y}_{k+1}^T)(\lambda^{-1}\mathbf{C}_k - \frac{\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1}\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k}{\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1} + 1}) \\ = \mathbf{B}_k\mathbf{C}_k - \lambda\mathbf{B}_k\frac{\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1}\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k}{\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1} + 1} + \\ \mathbf{z}_{k+1}\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k - \mathbf{z}_{k+1}\mathbf{y}_{k+1}^T\frac{\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1}\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k}{\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1} + 1}\end{aligned} \tag{15}$$

And then updating the dictionary can be written as

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k + \delta_k(\mathbf{z}_{k+1} - \mathbf{\Omega}_k\mathbf{y}_{k+1}){\mathbf{u}_k}^T \tag{16}$$

Where

$$\delta_k = \frac{1}{\mathbf{y}_{k+1}^T\lambda^{-1}\mathbf{C}_k\mathbf{y}_{k+1} + 1} \tag{17}$$

$$\mathbf{u_k} = (\lambda^{-1}\mathbf{C}_k)\mathbf{y}_{k+1} \tag{18}$$

$$\mathbf{u_k}^T = \mathbf{y}_{k+1}^T(\lambda^{-1}\mathbf{C}_k) \tag{19}$$

2.2.3. Practical Implementation

In the proposed RLS-ADL algorithm, a randomly initialized dictionary may not give stable convergence, as observed in our numerical tests. Therefore, we randomly select R ($R \ll K$) vectors from the observed data as the training subset $\mathbf{Y}_R$ and use the K-plane clustering algorithm to estimate a dictionary as $\mathbf{\Omega}_0$. And then We obtain $\mathbf{Z}_R = \mathbf{\Omega}_0\mathbf{Y}_R$. Finally, $\mathbf{B}_0$ and $\mathbf{C}_0$ can be initialized:

$$\mathbf{B}_0 = \mathbf{Z}_R\mathbf{Y}_R^T \tag{20}$$

$$\mathbf{C}_0 = (\mathbf{Y}_R\mathbf{Y}_R^T)^{-1} \tag{21}$$

In the RLS-ADL algorithm, an adaptive forgetting factor λ has been included to improve the convergence. In practice, λ can be tuned adaptively by gradually increasing λ towards 1 and such that the algorithm remembers more and more about what is already learned during the learning process.

$$\lambda(k+1) = \min(\lambda(k) + \lambda_{adj}, \lambda_{max}) \tag{22}$$

where λ_{adj} is an adjustment constant, and λ_{max} is the maximum of $\lambda(k)$. The proposed algorithm is summarized in Algorithm 1. It is worth noting that, to further improve the performance of the proposed algorithm, we have incorporated a post-processing step to reduce the coherence between the atoms of $\mathbf{\Omega}_\Lambda$ (as defined in Section 2.1), i.e. $\mathbf{\Omega}_\Lambda{\mathbf{\Omega}_\Lambda}^T = I$. More specifically, if the inner product of the atoms in $\mathbf{\Omega}_\Lambda$ is smaller than 0.9, we consider them as uncorrelated. Otherwise, we randomly generate an atom to replace the original one. In each iteration, the rows of $\mathbf{\Omega}_\Lambda$ are normalized to reduce the scaling ambiguity.

Algorithm 1: The RLS-ADL algorithm

Input: The training set $\mathbf{Y}$, and the co-sparsity l.
Output: Dictionary $\mathbf{\Omega} = \mathbf{\Omega}_K$.
Initialization: Initialise $\mathbf{B}_0$ and $\mathbf{C}_0$ using (20) and (21) respectively as described in Section 2.2.3. Set the initial values for the parameters λ, λ_{adj}, λ_{max}, and the number of iterations Q. Let $\lambda(0) = \lambda$.
For $q = 1...Q$ **do**
For $k = 0, ..., K-1$ **do**

- Compute $\mathbf{z}_{k+1} = \mathbf{\Omega}_k\mathbf{y}_{k+1}$, and select l numbers of $\mathbf{z}_{k+1}$ which have the smallest values and set the l values being zero.
- Estimate the analysis dictionary using the equations (16).
- Update the forgetting factor $\lambda(k+1)$ using (22).

End
Set $\mathbf{\Omega}_0 = \mathbf{\Omega}_K$.
End

3. COMPUTER SIMULATION

We present a set of experiments to demonstrate the performance of the proposed RLS-ADL algorithm. In the first part, we present results of the proposed algorithm for synthetic data, demonstrating its convergence with respect to the iteration numbers and its ability for recovering the ground-truth dictionary given a set of training signals. In the second part, we evaluate the performance of the proposed algorithm for natural image denoising and its associated computational cost. The Analysis K-SVD [10] and the K-plane clustering [14] are used as baseline algorithms for performance comparison. For the baseline K-plane clustering algorithm [14], the full training data are used, as opposed to the small set of data used in the K-plane clustering for the initialisation of the proposed algorithm. In these experiments, we choose empirically $R = 200$, $\lambda_{adj} = 0.000025$, $\lambda_{max} = 1$, and the initial value of $\lambda = 0.99$.

3.1. Experiments on synthetic data

In this experiment we use the proposed method to recover a dictionary that was used to produce the set of training data, compare the recovery percentage of the atoms by the proposed algorithm with the two baseline methods. We used the same experimental protocol as in [10] for these three algorithms. The analysis dictionary $\mathbf{\Omega} \in R^{50\times25}$ was generated with random Gaussian entries, and the data set consists of $K = 50000$ signals each residing in a 4-dimensional subspace with both the noise-free setup and a noise setup ($\sigma = 0.04, SNR = 25dB$). If $\min_i(1 - \left|\hat{\mathbf{w}}_i^T \mathbf{w}_j\right|) < 0.01$, a row $\mathbf{w}_j^T$ in the true dictionary $\mathbf{\Omega}$ is regarded as recovered, where $\hat{\mathbf{w}}_i^T$ are the atoms of the trained dictionary.

The convergence curve of the RLS-ADL algorithm is presented in Figure 1. For recovering the dictionary, similar to [10, 14], we run 100 iterations in the two baseline algorithms, and five iterations for the RLS-ADL algorithm. The results of the recovery percentage and runtime are presented in Table 1. It can be observed that the recovery percentage of the proposed RLS-ADL algorithm is better than both baseline algorithms. The RLS-ADL algorithm is computationally faster than the Analysis K-SVD algorithm but slower than the K-plane clustering algorithm (Computer OS: Windows 7, CPU: Intel Core i5-2410M @ 2.30GHz, RAM: 2G).

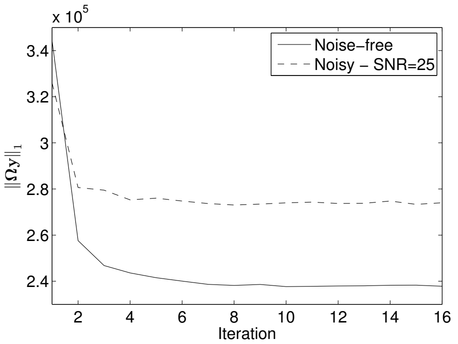

Fig. 1: The convergence curve of the RLS-ADL algorithm.

Table 1: Results of Recovery Percentage

Method	Noisy	Percentage	Runtime
Analysis K-SVD	free	0.9	11298s
	SNR=25	0.88	11341s
K-plane clustering	free	0.68	3487s
	SNR=25	0.44	3532s
RLS-ADL	free	0.96	5886s
	SNR=25	0.94	6101s

3.2. Experiments on image

In this section, we perform image denoising experiments based on the dictionaries learned by the proposed and the baseline algorithms. The test set consists of the three images commonly used in denoising (Lena, house and peppers) [10, 16]. The same experimental protocol is used for all the three algorithms. And we run 50 iterations in the two baseline algorithms, and one iteration for the RLS-ADL algorithm since the algorithm is convergent on image denoising. The dictionary of size 63×49 is created by using a training set of size 20,000 of 7×7 image patches. Noise of different levels σ, varying from 5 to 20, is added to these image patches. We also assume that the co-sparsity $l = M - 7$. In this experiment, if $\mathbf{\Omega}_\Lambda \mathbf{y}_k < 10^{-8}$, we can remove columns $\mathbf{y}_k$ from the matrix $\mathbf{Y}$. We use the OBG algorithm [10] to recover each image patch. The denoising performance is evaluated by the peak signal to noise ratio (PSNR) defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2 \times M \times K}{\Sigma_{i=1}^{M}\Sigma_{k=1}^{K}(\mathbf{y}_{ik} - \mathbf{x}_{ik})^2}\right) \qquad (23)$$

The denoising and runtime results are presented in Tables 2 and 3 respectively. We can observe that the performance of the proposed RLS-ADL algorithm is in general very similar to the baseline algorithms. The proposed algorithm, however, shows advantages in runtime over the baselines.

4. CONCLUSION

In this paper, the RLS-ADL algorithm has been proposed for analysis dictionary learning, directly exploiting the observed data without pre-estimating the signal $\mathbf{X}$ from its noisy version $\mathbf{Y}$. And we use the K-plane clustering algorithm with randomly selecting a small number of training vectors to obtain the initial dictionary. As a result, the proposed algorithm has a good convergence property and can also be used for online dictionary learning. The simulation experiments have shown its computational advantages over the baselines.

Table 2: Image denoising results (PSNR in dB)

σ	Noisy	Method	Lena	House	Peppers
5	34.15	Analysis K-SVD	38.43	39.20	37.89
		K-plane clustering	38.20	38.68	37.23
		RLS-ADL	38.06	38.23	37.37
10	28.13	Analysis K-SVD	34.85	35.27	33.80
		K-plane clustering	34.82	35.06	33.34
		RLS-ADL	34.39	34.54	33.34
15	24.61	Analysis K-SVD	32.59	33.00	31.31
		K-plane clustering	33.02	33.00	31.14
		RLS-ADL	32.37	32.45	31.00
20	22.11	Analysis K-SVD	31.42	31.49	29.80
		K-plane clustering	31.52	31.57	29.47
		RLS-ADL	31.17	31.10	29.48

Table 3: Dictionary learning runtime

Method	Lena	House	Peppers
Analysis K-SVD	8187s	7613s	8023s
K-plane clustering	4038s	3100s	3620s
RLS-ADL	598s	567s	581s

5. 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).

6. REFERENCES

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

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

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

[4] S.T. Li, H.T. Yin, and L.Y. Fang, "Remote sensing image fusion via sparse representations over learned dictionaries," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 9, pp. 4779–4789, 2013.

[5] T. Hu, J. Nunez-Iglesias, and S. Vitaladevuni, "Electron microscopy reconstruction of brain structure using sparse representations over learned dictionaries," *IEEE Transactions on Medical Imaging*, vol. 32, no. 12, pp. 2179–2188, 2013.

[6] E.T. Liu and B. Jafarpour, "Learning sparse geologic dictionaries from low-rank representations of facies connectivity for flow model calibration," *Water Resources Research*, vol. 49, no. 10, pp. 7088–7101, 2013.

[7] D. Damanik, A.M. Finkelshtein, and A. Iosevich, "3d data denoising using combined sparse dictionaries," *Mathematical Modelling of Natural Phenomena*, vol. 8, no. 1, pp. 60–74, 2013.

[8] H.J. Huang, J. Yu, and W.D. Sun, "Superresolution mapping using multiple dictionaries by sparse representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 12, pp. 2055–2059, 2014.

[9] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2121–2130, 2010.

[10] R. Rubinstein, T. Peleg, and M. Elad, "Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.

[11] Y. Zhang, H.L. Wang, T.L. Yu, and W. Wang, "Subset pursuit for analysis dictionary learning," in *The 21st European Signal Processing Conference*, 2013.

[12] B. Ophir, M. Elad, N. Bertin, M.D. Plumbley, and et al, "Sequential minimal eigenvalues-an approach to analysis dictionary learning," in *The 19th European Signal Processing Conference*, 2011, pp. 1465–1469.

[13] M. Yaghoobi, S. Nam, R. Gribonval, M.E. Davies, and et al, "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference*, 2011, pp. 1470–1474.

[14] Y. Zhang, H.L. Wang, W.Wang, and S. Sanei, "K-plane clustering algorithm for analysis dictionary learning," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2013.

[15] J.A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[16] S. Roth and M.J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.