OVERCOMPLETE SPARSIFYING TRANSFORM LEARNING ALGORITHM USING A CONSTRAINED LEAST SQUARES APPROACH

Ender M. Eksioglu and Ozden Bayir

Electronics and Communications Engineering Department
Istanbul Technical University
Istanbul, Turkey

ABSTRACT

Analysis sparsity and the accompanying analysis operator learning problem provide an important framework for signal modeling. Very recently, sparsifying transform learning has been put forward as an effective and new formulation for the analysis operator learning problem. In this study, we develop a new sparsifying transform learning algorithm by using the uniform normalized tight frame constraint. The new algorithm bypasses the computationally expensive analysis sparse coding step of the standard analysis operator learning algorithms. The resulting minimization problem is solved by alternating between two steps. The first step is the operator update, which comprises a least squares solution followed by a projection, and the second step is the sparse code update realized by a simple thresholding procedure. Simulation results indicate that the proposed algorithm provides improved analysis operator recovery performance when compared to a recent analysis operator learning algorithm from the literature, which uses the same uniform normalized tight frame constraint.

***Index Terms*—** Analysis operator learning; sparsifying transform learning; dictionary learning; sparse coding

1. INTRODUCTION

Sparse regularization of inverse problems has gained considerable impetus during the last decade, following the groundbreaking progress in solving the synthesis based sparse representation problem [1]. The revelation that the elusive, NP-complete ℓ_0 pseudo-norm sparse representation problem can be convexly relaxed by the use of the more benign ℓ_1 norm [2], has led to an outbreak of new sparse representation methods accompanied by advances in related fields such as compressive sensing [3] and dictionary learning. Dictionary learning deals with the problem of finding a proper set of synthesis atoms which facilitate sparse representation for a group of signal under scrutiny [4,5]. Various algorithms such as the original, synthesis K-SVD [6] have been proposed for learning an overcomplete dictionary appropriate for use in sparse representation from the signal corpus itself. The synthesis sparsity and dictionary learning have a lesser known counterpart in the recently introduced analysis sparsity framework [7, 8]. In analysis sparsity, the signal is assumed to be sparse in a transform domain defined over a suitable analysis operator. The analysis sparsity is also equivalently called as cosparsity, and various optimization based and greedy algorithms have been developed to solve the cosparse representation or cosparse coding problem [9, 10]. The cosparse representation problem has also been recast as a sparse representation problem, allowing the use of synthesis sparse representation algorithms for cosparse coding [11].

Similar to the dictionary learning algorithms used in synthesis sparsity approach, analysis operator learning algorithms for cosparse modeling of signals have been developed. Analysis K-SVD [12] extends the sequential and SVD-based update procedure of its well-known synthesis counterpart in [6] to the analysis operator learning problem. In [13] the authors utilize an optimization over manifolds approach to learn appropriate analysis operators. Another recent analysis operator learning algorithm is presented in [14]. In [14] the learned analysis operators are constrained to lie in the set of Uniformly Normalized Tight Frames, as to hinder possible degenerate solutions. After the conception of these analysis operator learning algorithms, a similar framework has been developed to determine operators which lead to analysis sparsity. This new framework as introduced in [15] has been dubbed as "Sparsifying Transform Learning". In sparsifying transform learning, the minimization problem for operator learning is formulated in a modified manner when compared to the minimization problems of the above listed algorithms. Nevertheless, this modification in the minimization formulation leads to the replacement of the expensive cosparse coding step of the conventional analysis operator learning algorithms with a thresholding step of much reduced complexity. The sparsifying transform learning framework has been utilized together with the K-SVD approach of [12] to formulate a new algorithm called as Transform K-SVD in [16].

In this work, we develop a new sparsifying transform learning algorithm by merging the transform learning approach of [15] with the constrained Analysis Operator

Learning (AOL) algorithm of [14]. We will call the newly developed transform learning algorithm as the Constrained Least Squares Sparsifying Transform Learning (CLS-TL) algorithm. We will compare the operator learning performance of this new algorithm with the AOL algorithm. Despite its reduced complexity, the new transform learning algorithm has comparable and even better performance when compared to the AOL algorithm. The rest of the paper is structured as follows. We first give a constrained formulation for the analysis operator learning approach. Next, we introduce the transform learning problem, and we develop a new constrained minimization formulation for transform learning. We continue with the development of a heuristic, iterative minimization algorithm for the solution of this new constrained transform learning problem. The simulations section details the exact operator recovery performance of this new transform learning algorithm vis-à-vis the AOL algorithm of [14]. We wrap up with the conclusions section.

2. CONSTRAINED ANALYSIS OPERATOR LEARNING

After the proliferation of synthesis sparsity based regularization for various inverse problems, dictionary learning has become a popular research interest [4–6]. Learning a specific synthesis dictionary for the actual data on hand, results in performance improvement when compared to the use of non-specific, analytic dictionaries which are generated without referencing the actual data. Dictionary learning can be formalized in the following form, where dictionary $\mathbf{D}$ is learned as to allow sparse representation of the data.

$$\min_{\mathbf{D}\in\mathscr{D},\mathbf{X}} \|\mathbf{D}\mathbf{X}-\mathbf{Y}\|_F^2, \text{ s.t. } \|\boldsymbol{x}_n\|_0 \le s\; \forall n=1,\dots,N \quad (1)$$

Here, $\mathbf{Y} \in \mathbb{R}^{M\times N}$ is the complete data matrix, where its columns $\boldsymbol{y}_n \in \mathbb{R}^M$ are the individual signal vectors for $n = 1,\dots,N$. $\mathbf{D} \in \mathbb{R}^{M\times K}$ is the synthesis dictionary with columns (also called as atoms) $\boldsymbol{d}_k \in \mathbb{R}^M$ for $k = 1,\dots,K$. The set $\mathscr{D}$ provides an admissability constraint for $\mathbf{D}$, where one usual choice is forcing uniformly normalized atoms, that is $\mathscr{D} = \big\{\mathbf{D} : \|\boldsymbol{d}_k\|_2 = 1, \forall k = 1,\dots,K\big\}$. The vectors $\boldsymbol{x}_n \in \mathbb{R}^K$ are the sparse representation vectors corresponding to $\boldsymbol{y}_n$, and they form the columns of $\mathbf{X} \in R^{K\times N}$. The dictionary learning problem as formalized above attempts to find a suitable dictionary which allows sparse synthesis of the given data family using its atoms.

Recently another type of sparsity structure has come under scrutiny. This is the analysis sparsity approach, which has also been called as cosparsity. In this case, the signal is assumed to be sparse in a transform domain, where the transformation is achieved through a suitable operator. The cosparsity of a given signal $\boldsymbol{y} \in \mathbb{R}^M$ with respect to an operator $\boldsymbol{\Omega} \in \mathbb{R}^{K\times M}$ is given by the cardinality of its co-support with respect to $\boldsymbol{\Omega}$. Co-support Λ of signal $\boldsymbol{y}$ is defined as Λ=$\big\{m : \{\boldsymbol{\Omega}\boldsymbol{y}\}_m = 0, \forall m = 1,\dots,M\big\}$ [14], that is the set of indices where $\boldsymbol{\Omega}\boldsymbol{y}$ becomes zero. By using (1) as inspiration, a noisy formulation of learning a suitable analysis operator for a given signal set can been given as follows:

$$\min_{\boldsymbol{\Omega}\in\mathscr{C},\mathbf{X}} \|\mathbf{X}-\mathbf{Y}\|_F^2, \text{ s.t. } \|\boldsymbol{\Omega}\boldsymbol{x}_n\|_0 \le s\; \forall n=1,\dots,N \quad (2)$$

In (2), $\mathscr{C}$ denotes an appropriate admissability set for constraining the learned analysis operator. The set $\mathscr{C}$ should be defined as to evade degenerate learned transforms such as those with repeated or all zero rows. As an example, in [12] where the Analysis K-SVD algorithm is developed, the analysis operator learning goal is similar to (2). The main minimization problem for operator learning presented in [12] is of the same form as (2), with $\mathscr{C}$ defined as follows:

$$\mathscr{C} = \big\{\boldsymbol{\Omega} :\ \text{rank}(\boldsymbol{\Omega}_{\Lambda_n}) = M - s, \forall n = 1,\dots,N, \text{ and } \|\boldsymbol{\omega}^k\|_2 = 1, \forall k = 1,\dots,K\big\}. \quad (3)$$

In the above equation, Λ_n denotes the co-support of the signal $\boldsymbol{y}_n$. The matrix $\boldsymbol{\Omega}_\Lambda$ is a sub-matrix of $\boldsymbol{\Omega}$ including the rows indexed in Λ. The row vector $\boldsymbol{\omega}^k$ is the k^{th} row of $\boldsymbol{\Omega}$. An equation similar to (2) is used in [14] for the analysis operator learning problem. The formulation in [14] convexly relaxes the learning problem by using the ℓ_1 norm instead of the ℓ_0 norm, and it also includes a Lagrangian multiplier as shown below:

$$\min_{\boldsymbol{\Omega}\in\mathscr{C},\mathbf{X}} \frac{\lambda}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \|\boldsymbol{\Omega}\mathbf{X}\|_1. \quad (4)$$

Here, a rather unconventional notation is used as $\|\cdot\|_1$ is taken to denote the sum of absolute values of the argument matrix entries [14]. In (4), $\mathscr{C}$ is defined to be the Uniform Normalized Tight Frame (UNTF) constraint, which was initially introduced in [17]. The UNTF constraint is a culmination of row norm and full rank constraints, and it is given as follows:

$$\mathscr{C} = \big\{\boldsymbol{\Omega} :\ \boldsymbol{\Omega}^T\boldsymbol{\Omega} = \mathbf{I}, \text{ and } \|\boldsymbol{\omega}^k\|_2 = 1, \forall k\big\}. \quad (5)$$

In (5), $\mathbf{I}$ denotes the identity operator. The AOL algorithm as proposed in [14] is based on a two-stage alternating minimization solution for (4). The two distinct steps of minimization over a single variable are solved individually.

$$\boldsymbol{\Omega}^{[i]} = \arg\min_{\boldsymbol{\Omega}\in\mathscr{C}} \|\boldsymbol{\Omega}\mathbf{X}^{[i-1]}\|_1 \quad (6a)$$

$$\mathbf{X}^{[i]} = \arg\min_{\mathbf{X}} \frac{\lambda}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \|\boldsymbol{\Omega}^{[i]}\mathbf{X}\|_1 \quad (6b)$$

In [14], the first part (6a) is solved by a subgradient descent step succeeded by an approximate projection onto the UNTF set. The second step in (6b) requires the solution of an analysis sparse coding problem for all the columns in $\mathbf{Y}$, which is computationally expensive compared to the first part. In the coming chapter we reformulate the constrained analysis operator learning problem as a transform learning problem.

3. CONSTRAINED SPARSIFYING TRANSFORM LEARNING

Sparsifying transform learning has been introduced in [15] as a more general paradigm for analysis operator learning. Transform learning avoids the use of the expensive sparse coding step in the operator/dictionary learning approaches, and hence promises to provide comparable operator learning performance at a much reduced cost [16]. Using both the sparsifying transform learning paradigm [15] and the constrained analysis operator learning problem from (4), we now present a new constrained formulation for the transform learning problem.

$$\min_{\mathbf{\Omega}\in\mathscr{C},\mathbf{X}} \|\mathbf{\Omega}\mathbf{Y}-\mathbf{X}\|_F^2+\eta\|\mathbf{X}\|_1 \tag{7}$$

We assume $\mathscr{C}$ to be the UNTF constraint of (5). Now, we will outline an algorithm for the solution of the novel minimization problem in (7). We adopt the two-step iterative approach as used in the algorithms from the literature. Hence, we seek to minimize the below given two goals at each iteration of an iterative algorithm.

$$\mathbf{\Omega}^{[i]}=\arg\min_{\mathbf{\Omega}\in\mathscr{C}}\|\mathbf{\Omega}\mathbf{Y}-\mathbf{X}^{[i-1]}\|_F^2 \tag{8a}$$

$$\mathbf{X}^{[i]}=\arg\min_{\mathbf{X}}\|\mathbf{\Omega}^{[i]}\mathbf{Y}-\mathbf{X}\|_F^2+\eta\|\mathbf{X}\|_1 \tag{8b}$$

The problem in (8b) corresponds to the analysis sparse coding step (6b). However, (8b) is simply solved by soft thresholding $\mathbf{\Omega}^{[i]}\mathbf{Y}$ as shown below [15]:

$$(\mathbf{X}^{[i]})_{k,n}=\begin{cases}(\mathbf{\Omega}^{[i]}\mathbf{Y})_{k,n}-\frac{\eta}{2}, & (\mathbf{\Omega}^{[i]}\mathbf{Y})_{k,n}\geq\frac{\eta}{2}\\ (\mathbf{\Omega}^{[i]}\mathbf{Y})_{k,n}+\frac{\eta}{2}, & (\mathbf{\Omega}^{[i]}\mathbf{Y})_{k,n}<-\frac{\eta}{2}\\ 0, & \text{else}\end{cases}. \tag{9}$$

Here, $(\cdot)_{k,n}$ denotes indexed matrix entries [15]. This exact solution in (9) is much simpler to obtain than solving (6b). The transform learning problem in (7) could have been formulated using an ℓ_0 sparsity condition as in (2). In this case, (8b) would transform into a problem which is exactly solved by hard thresholding [15]. In this paper, we have have used the convexly relaxed (8b) with the exact solution given in (9).

For the problem (8a), we propose the approximate solution of finding the least squares solution followed by a projection onto the UNTF set. The least squares solution is given by

$$\mathbf{\Omega}_{\text{ls}}^{[i]}=\mathbf{X}^{[i-1]}\mathbf{Y}^{\dagger}=\mathbf{X}^{[i-1]}\mathbf{Y}^T\big(\mathbf{Y}\mathbf{Y}^T\big)^{-1}. \tag{10}$$

We should note that the pseudo-inverse $\mathbf{Y}^{\dagger}$ stays constant throughout the iterations, hence it suffices to calculate it only once at the start. The final result is obtained by an approximate projection of $\mathbf{\Omega}_{\text{ls}}^{[i]}$ onto the UNTF:

$$\mathbf{\Omega}^{[i]}=\boldsymbol{\mathcal{P}}_{\text{UN}}\{\boldsymbol{\mathcal{P}}_{\text{TF}}\{\mathbf{\Omega}_{\text{ls}}^{[i]}\}\}. \tag{11}$$

Algorithm 1 Constrained Least Squares Sparsifying Transform Learning (CLS-TL)

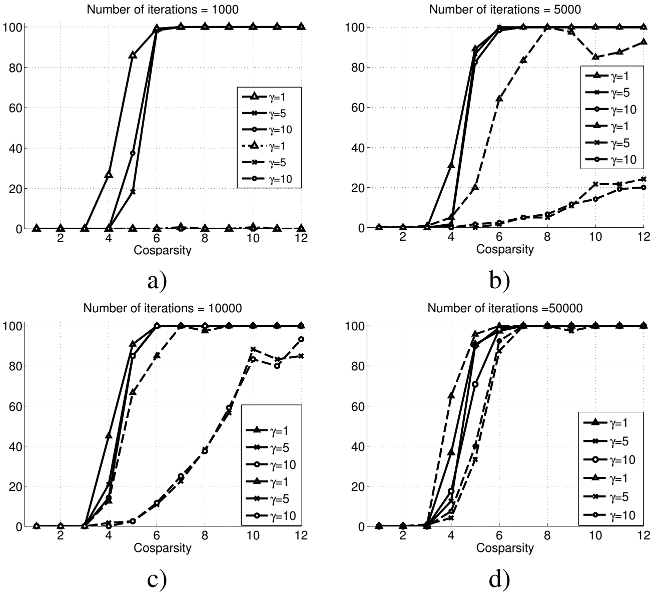
```
Input: Data record of length $N$, $\mathbf{Y}=\{\boldsymbol{y}_n\}_{n=1}^N$. Regularization constant $\eta$.
Goal: $\min_{\mathbf{\Omega}\in\mathscr{C},\mathbf{X}} \|\mathbf{\Omega}\mathbf{Y}-\mathbf{X}\|_F^2+\eta\|\mathbf{X}\|_1$

Initialize $\mathbf{\Omega}^{[0]}$ and calculate $\mathbf{X}^{[0]}=\lfloor\mathbf{\Omega}^{[0]}\mathbf{Y}\rceil_\eta$.
Calculate $\mathbf{Y}^{\dagger}=\mathbf{Y}^T\big(\mathbf{Y}\mathbf{Y}^T\big)^{-1}$.
for $i:=1,2,\ldots$ do                                          ▷ main iteration
    $\mathbf{\Omega}^{[i]}=\boldsymbol{\mathcal{P}}_{\text{UN}}\{\boldsymbol{\mathcal{P}}_{\text{TF}}\{\mathbf{X}^{[i-1]}\mathbf{Y}^{\dagger}\}\}$   ▷ Transform update step, complete with LS solution and UNTF projection.
    $\mathbf{X}^{[i]}=\lfloor\mathbf{\Omega}^{[i]}\mathbf{Y}\rceil_\eta$   ▷ transform sparse coding step realized by soft thresholding.
end for                                                         ▷ end of main iteration
```

Here, $\boldsymbol{\mathcal{P}}_{\text{UN}}$ denotes the projection onto the space of unit row norm (UN) frames, and as stated in [14] this can be simply obtained through scaling the rows. $\boldsymbol{\mathcal{P}}_{\text{TF}}$ on the other hand, is a projection onto the tight frame (TF) manifold, and this can be realized by calculating a singular value decomposition (SVD) of the argument. If $\mathbf{\Omega}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is an SVD, then $\boldsymbol{\mathcal{P}}_{\text{TF}}\{\mathbf{\Omega}\}=\mathbf{U}\,\mathbf{I}_{K\times M}\mathbf{V}^T$ is the required projection [14]. We call the above outlined novel practical algorithm for learning a sparsifying transform as the Constrained Least Squares Sparsifying Transform Learning (CLS-TL) algorithm. The CLS-TL algorithm is summarized in Alg.1. In Alg.1, $\lfloor\cdot\rceil_\eta$ denotes the soft thresholding of the argument matrix as given in (9).

4. SIMULATION RESULTS

In this section we present experimental results for the exact recovery of reference analysis operators using the proposed CLS-TL algorithm and the AOL algorithm of [14]. This exact operator recovery setup is adapted from [14]. Firstly, a particular reference analysis operator $\mathbf{\Omega}_0\in\mathbb{R}^{24\times16}$ is generated. The $\mathbf{\Omega}_0$ operator is obtained by repeatedly projecting an initial $\mathbf{\Omega}_{0-}\in\mathbb{R}^{24\times16}$ operator onto the UN and TF sets. The initial $\mathbf{\Omega}_{0-}$ operator is composed of iid, normal distributed, zero mean and unit variance elements. Next, a training signal set comprised of signals $\boldsymbol{y}_i, i=1\ldots l$ is constructed. Each signal $\boldsymbol{y}_i$ is generated as to have cosparsity q with respect to the reference analysis operator $\mathbf{\Omega}_0$. This is realized by randomly picking q rows from $\mathbf{\Omega}_0$, and then again randomly generating a vector from the orthogonal complement space of these chosen rows [14]. There is no observation noise added to these cosparse signals. For this noiseless setting the AOL algorithm reduces to noiseless AOL algorithm [14, 17] which repeatedly solves (6a). The noiseless AOL and CLS-TL have similar computational complexities, since the expensive sparse coding step (6b) of AOL is avoided. We have observed that for the simulations here, AOL takes approximately 1.5 times the computation time of CLS-TL. However, for



a)

b)

c)

d)

Fig. 1: Average percentage of analysis operator recovery versus cosparsity for different iteration numbers. The solid lines correspond to the introduced CLS-TL algorithm, and the dashed lines correspond to AOL of [14]. a) 1000 iterations, b) 5000 iterations, c) 10000 iterations, d) 50000 iterations.

a general noisy formulation, CLS-TL would have a more significant reduction in complexity when compared to the AOL.

The initial operator for the algorithms is created by repeatedly projecting a matrix $\boldsymbol{\Omega}_{\text{in}} = \boldsymbol{\Omega}_0 + \gamma \mathbf{N}$ onto the UN and TF manifolds. Here, $\mathbf{N}$ is a normalized random matrix, and the constant γ determines the deviation of the initial operator estimate from the true operator. A row of the original operator $\boldsymbol{\Omega}_0$ is assumed to be exactly recovered, if in the learned operator there is a row with at most $\sqrt{0.001}$ ℓ_2 distance from this particular row. The main performance index is the percentage of exactly recovered rows versus the cosparsity q, and for each point in the plots the setup is averaged over 100 independent trials. The cosparsity q changes as $q = 1, 2, \ldots, 12$, whereas a constant $\eta = 0.2$ is used for CLS-TL.

In the first experiment we choose the size of the training data set as $l = 768$. The setup is repeated by changing the total number of iterations as 1000, 5000, 10000 and 50000, and by choosing $\gamma = 1, 5$ and 10. The results are presented in Fig.1. Fig.1 indicates that the new CLS-TL algorithm converges quicker than the AOL algorithm, by giving exact recovery results for the short iteration lengths of 1000 and 5000. In general the CLS-TL algorithm can exactly recover the reference operator for lower values values of q when compared to the AOL. For higher values of γ, that is when the initial operator is farther from the ground truth operator, CLS-TL again has better recovery results than AOL.

In the second experiment set, we study the effect of the training data set size on the average operator recovery performance. We change the data corpus size as $l = 384, 768$ and 1536. We again consider the deviation parameter values $\gamma = 1, 5$ and 10. Fig.2 details the results for these experiment set. As we can see from Fig.2, CLS-TL outperforms the AOL algorithm except for $\gamma = 1$. For $\gamma = 5$ and 10, the CLS-TL algorithm has better recovery results for all three data set sizes. Looking at Fig.2, we can infer that CLS-TL is less sensitive with respect to operator initializations which are far from the ground truth. We can also state that both algorithms benefit from an increase in the training data size. As more training data becomes available, the reference operator is recovered exactly for lower cosparsity values.

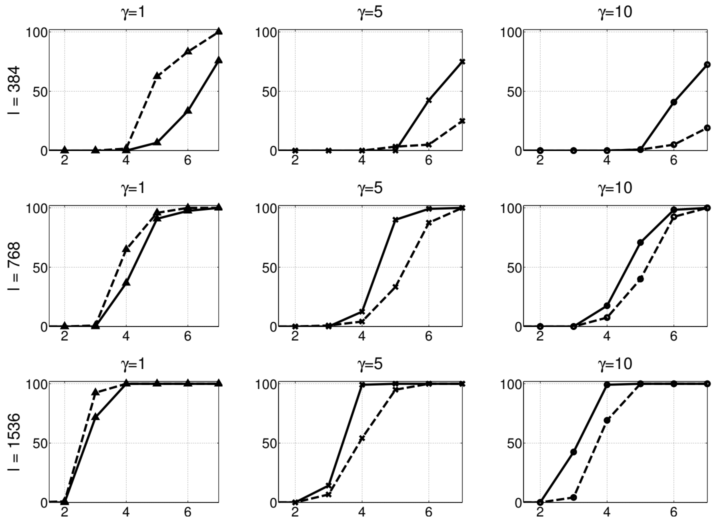


Fig. 2: Average percentage of analysis operator recovery versus cosparsity for different training data set sizes l. Number of iterations stays constant (50000). Solid lines are for the CLS-TL algorithm, and dashed lines are for the AOL of [14].

5. CONCLUSIONS

We have presented a constrained sparsifying transform learning framework, and we have developed a new transform learning algorithm by using this framework together with the UNTF set, which provides a viable constraint. We call the new algorithm as the Constrained Least Squares Sparsifying Transform Learning (CLS-TL) algorithm. We have compared the analysis operator recovery performance of the CLS-TL algorithm with the constrained AOL algorithm of [14]. In this exact analysis operator recovery setup, the CLS-TL algorithm has in general better performance at learning the underlying reference analysis operator. CLS-TL is able to recover the original analysis operator at lower cosparsity values when compared to the AOL algorithm. Hence, we can state the new CLS-TL algorithm provides a new and successful implementation of the transform learning approach for analysis operator learning.

6. REFERENCES

[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing.*, Springer, New York, NY, 2010.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[3] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Birkhäuser, New York, NY, 2013.

[4] M. D. Plumbley, "Dictionary learning for l1-exact sparse coding," in *ICA'07: Proceedings of the 7th international conference on Independent component analysis and signal separation*, Berlin, Heidelberg, 2007, pp. 406–413, Springer-Verlag.

[5] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2178–2191, 2009.

[6] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[7] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[8] S. Vaiter, G. Peyré, C. Dossal, and J. Fadili, "Robust sparse analysis regularization," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2001–2016, 2013.

[9] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[10] R. Giryes, S. Nam, M. Elad, R. Gribonval, and M.E. Davies, "Greedy-like algorithms for the cosparse analysis model," *Linear Algebra and its Applications*, vol. 441, pp. 22–60, 2014, Special Issue on Sparse Approximate Solution of Linear Systems.

[11] N. Cleju, M. G. Jafari, and M. D. Plumbley, "Analysis-based sparse reconstruction with synthesis-based solvers," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 5401–5404.

[12] 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, 2013.

[13] 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, 2013.

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

[15] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1072–1086, 2013.

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

[17] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference (EUSIPCO'11)*, 2011.