AN INITIALIZATION STRATEGY FOR THE DICTIONARY LEARNING PROBLEM

Cristian Rusu ⋆ *Bogdan Dumitrescu*†

⋆ IMT Institute for Advanced Studies Lucca, Italy
† Department of Automatic Control and Computers, University Politehnica of Bucharest, Romania

ABSTRACT

In this paper we present an efficient initialization strategy that improves the performance of overcomplete dictionary learning algorithms. The procedure exploits incoherent structures that can be manipulated and adapted to a given dataset relatively fast. The algorithm involves an iterative adaptation of the dictionary to the dataset with pruning of the less used atoms and constructions of new atoms that fit the data better. Experimental simulations show that the proposed method improves the performance of classical and new developments in dictionary learning algorithms.

Index Terms— sparse representations, dictionary learning, initialization.

I. INTRODUCTION

Problem. We investigate the construction of overcomplete dictionaries based on training data, also called dictionary learning, which is of great interest in the signal processing community [1]. Given a dataset $\boldsymbol{Y} \in \mathbb{R}^{n\times N}$ and a target sparsity s (maximum number of atoms allowed in each representation) the problem is to create dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ and the sparse representations matrix $\boldsymbol{X} \in \mathbb{R}^{m\times N}$ such that $\boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{X}$. The problem can be formulated as:

$$\begin{aligned} \underset{\boldsymbol{D},\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F \\ \text{subject to} \quad & \|\boldsymbol{x}_i\|_0 \le s,\ 1 \le i \le N \\ & \|\boldsymbol{d}_j\|_2 = 1,\ 1 \le j \le m, \end{aligned} \tag{1}$$

where $\|\boldsymbol{x}_i\|_0$ is the ℓ_0 pseudo-norm (the number of non-zero components in column $\boldsymbol{x}_i$), $\|\boldsymbol{E}\|_F^2 = \sum_i \sum_j e_{ij}^2$ is the Frobenius norm and the columns $\boldsymbol{d}_j$ of the dictionary $\boldsymbol{D}$, called atoms, are normalized. Most popular solutions to (1) involve an alternating optimization process:

- Keep dictionary $\boldsymbol{D}$ fixed and optimize the sparse representations $\boldsymbol{X}$ by using an approximate sparse reconstruction algorithm (e.g. OMP [2], ℓ_1 [3]).
- Keep the representations matrix $\boldsymbol{X}$ fixed and update the dictionary $\boldsymbol{D}$. Two popular update methods include MOD [4] and K–SVD [5] (AK–SVD [6]).

The best performance (higher convergence rate and lower running time) is achieved by the AK–SVD (Approximate K–SVD) algorithm which uses a Batch–OMP algorithm, a power method approximation to the SVD steps and a dictionary step that jointly updates an atom and all its sparse representations.

Overall, overcomplete dictionaries are characterized, among other, by the following important properties: representation error $\epsilon = \|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F$, mutual coherence $\mu(\boldsymbol{D}) = \max_{i<j} |\boldsymbol{d}_i^T \boldsymbol{d}_j|$ and dictionary dimension m.

Contribution. In this paper we focus on constructing dictionaries with low representation error. The main idea of the paper is to use incoherent structures to create a very good initialization for a dictionary learning algorithm like AK–SVD for example.

The benefits of constructing an initial dictionary have to overcome the cost of its construction. The proposed initialization procedure should be very fast, so that extra iterations of AK–SVD are not able to converge to similar results in the same amount of time. Ideally, it should also converge to representation error levels lower than the ones of AK–SVD even when a good, well known, initialization is provided.

The proposed method can also be used to initialize class models in a supervised classification context where dictionaries apply [7] [8]. Here, the availability of a well known initialization point (the same for each class) may affect negatively the discriminative/separation power of the models.

To achieve these desirable properties, the initial dictionary is learnt using rotation transforms on a given highly incoherent initial frame. The choice of incoherent frames is made such that a large variety of directions is explored right from the beginning. As the algorithm progresses, some of the unused (or less used) directions are removed and replaced with new atoms that give a significant decrease in the representation error. The assumption that not all atoms in an incoherent frame are equally useful is not restrictive since real world data should be highly correlated.

Relation with prior work. This seems to be the first time an initialization algorithm is proposed for the overcomplete dictionary learning problem. The review paper [9] does not even mention initialization. Until now, most learning procedures would either use a random initial dictionary, an overcomplete wavelet/Fourier dictionary or a sample of data items from $\boldsymbol{Y}$. Of course, we compare these classic strategies with the newly proposed initialization algorithm.

Rotation transforms (accompanied also by projections) have been use in the past for general dictionary learning to balance the trade-off between representation performance

C. Rusu (e-mail: cristian.rusu@imtlucca.it) is the corresponding author.

B. Dumitrescu's (e-mail: bogdan.dumitrescu@acse.pub.ro) work was supported by Romanian National Authority for Scientific Research, CNCS UEFISCDI, under Project PN-II-ID-PCE-2011-3-0400.

and mutual coherence [10]. The idea of (re)growing a full dictionary from a smaller one is inspired by [11]; however, here the added atoms are not retrained immediately, hence the expansion is very quick.

Contents. The paper is organized as follows: Section 2 presents the initialization algorithm, Section 3 describes numerous runs and comparisons using state-of-the-art dictionary learning algorithm and Section 4 concludes the paper.

II. INITIAL DICTIONARY CONSTRUCTION

In this section we describe an initialization scheme for overcomplete dictionary learning dictionaries based on incoherent dictionary learning methods.

The algorithm assumes the availability of an incoherent frame (either overcomplete or square – orthogonal) that serves as the initial point for a dictionary learning algorithm. We then adapt the frame to the dataset that is provided. We use the word adapt because the frame only suffers fast rotational transformations. Together with the rotations we also prune unused atoms from the frame. The idea described in this paper is to prune under-utilized, or unused, atoms in the initialization phase and then add new atoms by looking at the worst constructed data items from the dataset using the SVD. The construction of the new atoms is based on ideas from [11]. After the initial dictionary is constructed, the AK–SVD algorithm is applied. By using our initialization strategy, that performs very well in terms of running time, the hope is that fewer iterations of AK–SVD are needed to reach the same representation errors.

In this section we focus on reaching very good representation performance for dictionaries of fixed length m without concern for the mutual coherence. The initialization procedure, called DIA, developed in two stages is presented in Algorithm 1. Step A of DIA is very fast due to the dictionary update step 2a that only applies a SVD and the fast sparse approximation algorithm, OMP. The highly incoherent frames $\boldsymbol{F}$ can be provided by the IDCO algorithm [12]. Alternatively, we can take $\boldsymbol{F}$ to be orthonormal.

Each iteration of Step A begins by constructing the new rotation $\boldsymbol{Q}_k$ that is applied to the current dictionary. Considering the representations matrix fixed, we use (3) to construct this rotation. At iteration k, the first step consists mostly of matrix multiplications (even more, $\boldsymbol{X}_{k-1}$ is sparse), since the SVD from (3) is applied on a small matrix $\boldsymbol{Y}\boldsymbol{X}_{k-1}^T\boldsymbol{D}_{k-1}^T \in \mathbb{R}^{n\times n}$, where $n \leq m \ll N$ and hence is very fast. Notice that, if the algorithm converges, the transforms $\boldsymbol{Q}_k$ tend to $\boldsymbol{I}_n$.

We mention here that we use the publicly available library OMP-box that contains an efficient implementation of Batch-OMP [6]. Within the library we use the fastest implementation (even though it is the most memory consuming), whose inputs are: the target sparsity s, the projections $\boldsymbol{D}_k^T\boldsymbol{Y}$ and the Gram matrix $\boldsymbol{D}_k^T\boldsymbol{D}_k$. Notice that the Gram matrix is invariant to rotation transformations ($\boldsymbol{D}_k^T\boldsymbol{D}_k = \boldsymbol{F}^T\boldsymbol{F}$). When an atom is pruned we also decrease the dimension of the Gram matrix by removing the associated row and column. In terms of speed, step 2c is by far the slowest. Over 90% of the running time is spent in this sparse approximation step. Moreover, the dimension of the problem decreases due to the pruning step 2d every 10 iterations. The atoms to be removed are selected based on a threshold applied to the sum of the representation coefficients squared. The pruning process depends on an internal threshold parameter T. Taking into account that the total energy content of the coefficients at iteration k is $S = \|\boldsymbol{X}_k\|_F$ the threshold should be set to a percentage of this value. Alternatively, a decision can be made based on the relative importance of the atoms among themselves. This pruning step is not applied at every iterations. In this implementation we prune atoms every 10 iterations to allow the algorithm to start its convergence and thus making sure that the low score atoms are not in a transitional stage.

Algorithm 1 Dictionary Initialization Algorithm (DIA). Given the dataset $\boldsymbol{Y}$, the target dimension of the dictionary m, the incoherent frame $\boldsymbol{F} \in \mathbb{R}^{n\times p}$, $p \leq m$, the number of iterations K, pruning threshold T, number of working atoms L, reconstruction percentage P and sparsity s construct $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ that significantly reduces $\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F$.

• **Step A.** Adapt the frame $\boldsymbol{F} \in \mathbb{R}^{n\times p}$ to the available dataset $\boldsymbol{Y}$ by rotational transforms with an additional pruning step to produce the dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times r}, r \leq m$.

1) Construct representations $\boldsymbol{X}_0 = \text{OMP}(\boldsymbol{Y}, \boldsymbol{F})$.
2) Iterations $k = 1, \ldots, K$
   a) With $\boldsymbol{D}_{k-1}$ and $\boldsymbol{X}_{k-1}$ fixed, find $\boldsymbol{Q}_k$ by solving the orthogonal Procrustes problem [13]:
   $$\begin{aligned} \underset{\boldsymbol{Q}_k}{\text{minimize}} \quad & \|\boldsymbol{Y} - \boldsymbol{Q}_k\boldsymbol{D}_{k-1}\boldsymbol{X}_{k-1}\|_F \\ \text{subject to} \quad & \boldsymbol{Q}_k\boldsymbol{Q}_k^T = \boldsymbol{I}_n, \end{aligned} \tag{2}$$
   whose solution is $\boldsymbol{Q}_k = \boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U}$, $\boldsymbol{V}$:
   $$\boldsymbol{Y}\boldsymbol{X}_{k-1}^T\boldsymbol{D}_{k-1}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{3}$$
   b) Update the dictionary $\boldsymbol{D}_k = \boldsymbol{Q}_k\boldsymbol{D}_{k-1}$.
   c) Construct representations $\boldsymbol{X}_k = \text{OMP}(\boldsymbol{Y}, \boldsymbol{D}_k)$.
   d) Eliminate atoms j with score:
   $$S_j = \sum_{i=1}^{N} \boldsymbol{X}_k(j,i)^2 < T. \tag{4}$$

• **Step B.** With the resulting $\boldsymbol{D} \in \mathbb{R}^{n\times r}$, with $r \leq m$, iteratively expand the dictionary until the total number of atoms becomes m. Iterative process:

1) Construct L new atoms using the SVD on the worst reconstructed $P\%$ data items indexed by:
   $$\mathbb{W} = \left\{ \frac{\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2}{\|\boldsymbol{y}_i\|_2^2} \right\}_{<1,\ldots,\lceil PN\rceil>}, i = 1, \ldots, N, \tag{5}$$
   where $\boldsymbol{z}_{<i>}$ stands for the index of the i^{th} smallest component of $\boldsymbol{z}$. Add new atoms to $\boldsymbol{D}$.
2) Construct new representations $\boldsymbol{X} = \text{OMP}(\boldsymbol{Y}, \boldsymbol{D})$.
3) Check if dimension of current dictionary exceeds m:
   - If it does, prune the extra atoms by (4) and stop.
   - Otherwise, continue iterations.

Table I: Running times of learning algorithms and their total representation errors, in 3 separate contexts for various s. Initialization of K–SVD and of DIA is the same incoherent, overcomplete random frame of size m.

m	K–SVD		DIA+K–SVD(p)		DIA+K–SVD	
	ϵ	t	ϵ	t	ϵ	t
$s = 4$						
100	32.2	12.1	32.1	1.2	31.0	11.7
128	29.9	13.7	29.9	3.3	29.4	13.4
192	27.5	17.8	27.5	6.4	27.0	17.3
256	25.9	21.4	25.9	12.3	25.6	21.5
$s = 8$						
100	23.7	24.3	23.7	1.7	22.6	24.4
128	22.0	27.2	22.0	2.3	21.0	27.9
192	19.5	34.0	19.4	7.8	19.1	34.5
256	18.2	40.7	18.1	13.0	17.8	41.8
$s = 16$						
100	17.2	62.6	16.3	1.9	14.7	61.9
128	15.0	67.2	15.0	2.5	13.6	67.1
192	12.4	79.9	12.4	6.6	11.7	81.1
256	11.3	93.5	11.3	11.4	10.6	95.8

Table II: Running times of learning algorithms and their total representation errors, in 3 separate contexts for various s. Initialization is a random subsample of the data for the K–SVD and a random orthonormal basis for the DIA.

m	K–SVD		DIA+K–SVD(p)		DIA+K–SVD	
	ϵ	t	ϵ	t	ϵ	t
$s = 4$						
100	33.0	11.5	32.7	0.9	31.1	11.6
128	30.5	13.3	30.5	1.7	29.1	13.3
192	28.5	17.1	28.4	2.5	26.8	17.3
256	26.6	20.7	26.6	4.9	25.2	21.5
$s = 8$						
100	25.2	24.7	23.2	0.8	21.4	24.3
128	23.3	26.9	22.3	0.9	20.1	27.2
192	20.5	32.9	20.3	2.9	18.3	34.7
256	19.5	39.0	19.2	4.8	17.2	42.5
$s = 16$						
100	17.2	62.2	14.0	1.8	12.5	62.3
128	15.5	67.5	13.3	2.2	11.6	67.0
192	13.3	81.1	13.0	5.1	10.8	82.6
256	12.4	93.5	12.1	7.9	10.1	97.5

The result of this step is a dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times r}$, with $r \leq m$, that is still highly incoherent but manages to provide a good representation of $\boldsymbol{Y}$.

Step B of DIA expands, in an iterative fashion, the pruned dictionary obtained from Step A. The new atoms that are added to the dictionary are computed by applying a singular value decomposition on a small subset of the data items that have the highest reconstruction errors in the old dictionary. Experimentally, it has been observed that the mutual coherence increases significantly even from the first iteration. This means that we no longer have useful incoherent directions to add to the current dictionary, all new atoms added will cause consistent decreases in the representation error.

The resulting dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ is used as starting point for the AK–SVD algorithm.

III. RESULTS

In this section we describe a set of results obtained by DIA in various settings.

Overall, we try three popular initialization strategies: random incoherent dictionary, random sample from the available dataset and the overcomplete DCT.

First, we consider a set of $N = 11000$ image patches (8×8) extracted from the publicly available dataset Yale-Faces [14]. Means are removed and data is scaled to unit energy. This training data is concatenated in the test matrix $\boldsymbol{Y} \in \mathbb{R}^{64 \times 11000}$. We design overcomplete dictionaries $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ with target sparsity s. The K–SVD[1] algorithm runs for $K_{\text{K–SVD}} = 100$ iterations. We use the fast AK–SVD (Approximate K–SVD) variant that performs very similarly to the K–SVD. AK–SVD uses a batch OMP implementation and an approximation of the SVD with the power method to greatly reduce the running time.

[1]http://www.cs.technion.ac.il/%7Eronrubin/software.html

Table III: Stagewise K–SVD results in the same experimental context as Tables I and II.

m	$s = 4$		$s = 8$		$s = 16$	
	ϵ	t	ϵ	t	ϵ	t
100	31.2	146	22.3	257	13.6	500
128	29.3	190	20.5	355	12.3	646
192	26.9	333	18.2	563	10.5	1085
256	25.4	507	17.0	828	9.6	1570

We begin by showing in Figure 1 an internal result regarding the evolution of the DIA when initialized with an incoherent frame of size $p = 320$ and a random orthonormal basis, for $K = 150$ and $K = 50$ iterations respectively. Of course, the pruning process is less effective for the smaller orthonormal frame. Parameters of DIA in all experiments: $P = 5\%$, $L = 20$ and $T = S/100$ where $S = \|\boldsymbol{X}_k\|_F$.

Generally, we are interested in performance indicators (total representation error ϵ and running time t, in seconds) for K–SVD with DIA start in the following contexts:

- DIA+K–SVD(p) – a partial run of K–SVD. The algorithm stops earlier than $K_{\text{K–SVD}}$ iterations if the representation error of K–SVD is achieved.
- DIA+K–SVD – a full run by K–SVD.

The simulation results are presented in Tables I and II. In these two contexts, the starting point is provided by DIA. The difference is that in one case (Table I) the initial frame is a highly incoherent, overcomplete frame of size m while in the second case the initial frame is a random orthonormal matrix. As expected, observe that within each

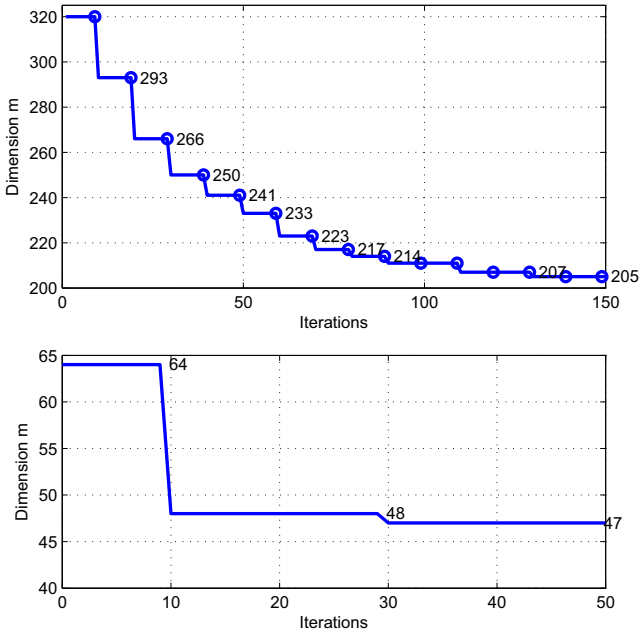

Fig. 1: Evolution of dictionary dimension with number of iterations for DIA for an initial highly incoherent overcomplete of size 320 (up). Same simulation context as previous plot for an initial orthonormal frame of size $m = 64$ (down).

table the representation error is smaller when using the DIA. For slightly lower representation error, the running time is reduced greatly while for the full run the process converges to lower error. Across Tables I and II the DIA outperforms the two initializations used for the K–SVD. The partial runs finish in general much sooner than the regular K–SVD. Interestingly, observe that representation errors for K–SVD are higher when initializing with a subset of the dataset.

To serve as reference, we also present the results obtained by using Stagewise K–SVD [11] in Table III with internal parameters $H = 3, R = 3$. This method does not need an initialization. It trains and builds the dictionary by increasing its dimension at every step so that it produces a significant decrease in the representation error. The drawback of such a method is the high running time. In terms of the representation performance the results are slightly better.

Considering the new developments in dictionary learning algorithms presented in [15] and named DT, we provide some comparative results by using the publicly available source code of this paper. We show that the initialization strategy proposed in this paper lowers the representation errors no matter what strategies are deployed in the dictionary update and coefficient computation steps (all of which can be seen as extensions of the basic K–SVD algorithm). We consider the Dictionary Update Cycles (DUC) method that consists of multiple dictionary updates step (in between each OMP step) and the new Coefficient Reuse OMP (CROMP) which is a variant of OMP that uses the coefficients computed in the previous step of K-SVD as a warm start for

Table IV: Comparison of DIA with dictionary learning strategies described in [15]. The initialization is done using a random sample of the dataset. Target sparsity is $s = 4$ and DIA starts with an orthonormal frame.

	m			
	100	128	192	256
DT, DUC = 1	22.26	21.06	19.42	18.38
DIA+DT, DUC = 1	21.37	20.24	18.90	17.84
DT, DUC = 4	22.07	21.09	19.41	18.36
DIA+DT, DUC = 4	21.25	20.19	18.81	17.80
DT + CROMP	19.95	18.49	17.50	16.81
DIA+DT+CROMP	18.36	17.56	16.60	15.96

Table V: Same context as Table IV but the initialization is done using an overcomplete DCT dictionary and DUC = 4.

	m			
	100	128	192	256
DCT+DT+CROMP	18.35	17.63	16.68	15.98
DIA+DT+CROMP	18.16	17.48	16.57	15.90

the new OMP application. This new simulation context is the one provided by [15]: the training data consisting of $N = 10000$, 8×8 patches extracted from popular test images (Lena, peppers, boat etc.), $s = 6$, the DC component removed and normalized. We compare the initialization strategy used (random selection of data items) to DIA. Results are presented in Table IV.

In the last experimental context, we utilize the DT training methods initialized with an overcomplete DCT (ODCT) frame. Since these dictionaries perform very well with image data we expect the results to be closer than in previous simulations. Of course, whenever good initial dictionaries are available they should be the first choice. Still, the results in Table V show that DIA is able to produce very good results, especially for larger dictionaries. In this last run, Step A reduces the initial frame $\boldsymbol{F}$ to produce a dictionary that has only, in increasing order of frame sizes m: 39, 43, 58 and 55 atoms. Step B then proceeds to complete the dictionaries to the full length m.

IV. CONCLUSIONS

In this paper we describe an initialization algorithm for the efficient construction of dictionaries used in the sparse overcomplete learning framework. The method is based on incoherent structures that are pruned and then adapted to the available dataset, while they span various directions of the working space. The results show that the method provides very good initial dictionaries that allow the learning methods to reach lower representation errors. All numerical experiments use the K–SVD algorithm (and variants) to prove the concept.

V. REFERENCES

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

[2] J. A. Tropp, Greed is good: Algorithmic results for sparse approximation, IEEE Trans. Inform. Theory, 50:2231–2242, 2004.

[3] D. L. Donoho, S. S. Chen and M. A. Saunders, Atomic decomposition by basis pursuit. SIAM J. Sci. Comput., 20:33–61, 1998.

[4] K. Engan, K. Skretting and J. Hakon-Husoy, Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation, Digital Signal Processing, 17(1):32–49, 2007.

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

[6] R. Rubinstein, M. Zibulevsky and M. Elad, Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit, CS Technical Report, Technion - Israel Institute of Technology, 2009.

[7] Z. Jiang, Z. Lin and L. S. Davis, Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD, IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[8] M. Liu, L. Lu, X. Ye, S. Yu and M. Salganicoff, Sparse classification for computer aided diagnosis using learned dictionaries, MICCAI, 6893:41–48, 2011.

[9] R. Rubinstein, A. M. Bruckstein and M. Elad, Dictionaries for sparse representation modeling, Proc. IEEE, 98(6):1045–1057, 2010.

[10] D. Barchiesi and M. D. Plumbley, Learning incoherent dictionaries for sparse approximation using iterative projections and rotations, IEEE Trans. on Signal Processing, 61(8):2055–2065, 2013.

[11] C. Rusu and B. Dumitrescu, Stagewise K-SVD to design efficient dictionaries for sparse representations, IEEE Signal Processing Letters, 19(10):631–634, 2012.

[12] C. Rusu, Design of incoherent frames via convex optimization, IEEE Signal Processing Letters, 20(7):673–676, 2013.

[13] P. Schonemann, A generalized solution of the orthogonal procrustes problem, Psychometrika, 31(1):1–10, 1966.

[14] Yale Face Database. [Online]. Available: http://cvc.yale.edu/projects/ yalefaces/yalefaces.html.

[15] L. N. Smith and M. Elad, Improving Dictionary Learning: Multiple Dictionary Updates and Coefficient Reuse, IEEE Signal Process. Letters, 20(1):79–82, 2013.