Orthogonal Procrustes analysis for dictionary learning in sparse representation

Giuliano Grossi, Raffaella Lanzarotti, and Jianyi Lin

***Abstract*—In the sparse representation model, the design of overcomplete dictionaries plays a key role for the effectiveness and applicability in different domains. Recent research has produced several dictionary learning approaches, being proven that dictionaries learnt by data examples significantly outperform structured ones, e.g. wavelet transforms. In this context, learning consists in adapting the dictionary atoms to a set of training signals in order to promote a sparse representation that minimizes the reconstruction error. Finding the best fitting dictionary remains a very difficult task, leaving the question still open. A well-established heuristic method for tackling this problem is an iterative alternating scheme, adopted for instance in the well-known K-SVD algorithm. Essentially, it consists in repeating two stages; the former promotes sparse coding of the training set and the latter adapts the dictionary to reduce the error. In this paper we present R-SVD, a new method that, while maintaining the alternating scheme, adopts the Orthogonal Procrustes analysis to update the dictionary atoms suitably arranged into subdictionaries. Comparative experiments on synthetic data and on the tasks of ECG compression and image modeling have been conducted, showing that on average R-SVD performs better than K-SVD.**

***Index Terms*—dictionary learning, sparse representation, Orthogonal Procrustes analysis, k-Means, ECG compression, image modeling**

I. Introduction

In many application domains, such as denoising, classification and compression of signals [1]–[3], it is often convenient to use a compact signal representation, following Occam's Razor principle. The classical linear algebra approach to this parsimonious representation is the sparsity: consider an overcomplete *dictionary* matrix $D \in \mathbb{R}^{n\times m}$ $(n < m)$ with columns $d_i, i = 1, ..., m$, called atoms, and a signal vector $y \in \mathbb{R}^n$; the sparsity approach consists in expressing y as linear combination $\sum_i x_i d_i$ with as few as possible non-zero coefficients $x_i \in \mathbb{R}$. Formally, the *sparse approximation* problem consists in finding $x \in \mathbb{R}^m$ minimizing the least squares error $\|y - Dx\|_2$ under the constraint that its ℓ_0-norm $\|x\|_0 := \#\{i : x_i \neq 0\}$ be at most a threshold $k \in \mathbb{N}$, i.e. x is *k-sparse*. This problem is combinatorial in nature and hence NP-hard [4]. Specifically, the sparse representation of a given set of probe signals poses two relevant questions.

The first concerns the development of efficient algorithms for solving the sparse approximation problem. To mention just a few, we recall those based on ℓ_0-minimization such as the greedy methods Matching Pursuit (MP) [5] and Orthogonal Matching Pursuit (OMP) [6], or those based on ℓ_1-minimization such as Basis Pursuit (BP) [7] and the Lasso [8].

The second issue, that we tackle in this article, concerns the design of suitable dictionaries that adaptively capture the model underlying the data. In literature, the proposed methods of *dictionary design* can be classified into two types [9]. The former consists in building *structured dictionaries* generated from analytic prototype signals. For instance, these comprise dictionaries formed by set of time-frequency atoms such as window Fourier frames and Wavelet frames [10], adaptive dictionaries based on DCT [11], Gabor functions [5], bandelets [12] and shearlets [13]. Within this category, it is worth mentioning the models introduced to capture the superposition of multiple phenomena by merging structured dictionaries [7], [14]. The latter type of design methods arises from the machine learning field and consists in *training a dictionary* from available signal examples, that turns out to be more adaptive and flexible for the considered data and task. The first approach in this sense [15] proposes a statistical model for natural image patches and searches for an overcomplete set of basis functions (dictionary atoms) maximizing the average log-likelihood (ML) of the model that best accounts for the images in terms of sparse, statistically independent components. In [16], instead of using the approximate ML estimate, a dictionary learning algorithm is developed for obtaining a Bayesian MAP-like estimate of the dictionary under Frobenius norm constraints. The use of Generalized Lloyd Algorithm for VQ codebook design suggested the iterative algorithm named MOD (Method of Optimal Directions) [17]. It adopts the alternating scheme, first proposed in [18], consisting in iterating two steps: signal sparse decomposition and dictionary update. In particular, MOD carries out the second step by adding a matrix of vector-directions to the actual dictionary. Alternatively to MOD, the methods that use least-squares solutions yield optimal dictionary updating, in terms of residual error minimization. For instance, such an optimization step is carried out either iteratively in ILS-DLA [19] on the whole training set (i.e. as batch), or recursively in RLS-LDA [20] on each training vector (i.e. continuously). In the latter method the residual error includes an exponential factor parameter for forgetting old training examples. With a different approach, K-SVD [2] updates the dictionary atom-by-atom while re-encoding the sparse non-null coefficients. This is accomplished through rank-1 singular value decomposition of the residual submatrix, accounting for all examples using the atom under consideration.

In this work we propose R-SVD (Rotate-SVD), an algorithm for dictionary learning in the sparsity model, inspired by a

The authors are with the Department of Computer Science, University of Milan, Via Comelico 39, 20135 Milano, Italy (e-mail: grossi@di.unimi.it, lanzarotti@di.unimi.it, jianyi.lin@unimi.it).

type of statistical shape analysis, called Procrustes[1] method [21], which has applications also in other fields such as psychometrics [22] and crystallography [23]. In fact, it consists in applying Euclidean transformations to a set of vectors (atoms in our case) to yield a new set with the goal of optimizing the model fitting measure. While maintaining the alternating scheme, R-SVD algorithm splits the dictionary into several sub-dictionaries and applies the Orthogonal Procrustes analysis simultaneously to all the atoms in each sub-dictionary. The technique is able to find an optimal dictionary after few iterations of the scheme. Notice that the proposed method differs from K-SVD [24], which instead updates one atom at a time together with the corresponding sparse coefficients. Several experimental sessions show that R-SVD is effective and behaves better than K-SVD in many synthetic conditions and in real-world applications.

In Sec. II we describe the problem and the proposed R-SVD algorithm. In Sec. III we first study the sub-dictionary size parameter of the method (sec. III-A), we show results on typical synthetic data experiments (sec. III-B), and we report applications to ECG signal compression (sec. III-C) and image modeling (sec. III-D). Finally, we draw some conclusions in Sec. IV.

II. Method

Suppose we are given the training dataset $Y = \{y_i\}_{i=1}^L \in \mathbb{R}^{n\times L}$. The dictionary learning problem consists in finding an overcomplete dictionary matrix $D = \{d_i\}_{i=1}^m \in \mathbb{R}^{n\times m}$ ($n < m$), which minimizes the least squares errors $\|y_i - Dx_i\|_2^2$, provided that all $x_i \in \mathbb{R}^m$ are k-sparse. By letting $X = \{x_i\}_{i=1}^L \in \mathbb{R}^{m\times L}$ denote the coefficient matrix, this problem can be formally stated as

$$\underset{D\in\mathbb{R}^{n\times m}, X\in\mathbb{R}^{m\times L}}{\operatorname{argmin}} \|Y-DX\|_F^2 \qquad \text{subject to} \qquad \|x_i\|_0 \le k, \quad i=1,\dots,L \tag{1}$$

Moreover, w.l.o.g. atoms in D are constrained to be unit ℓ_2-norm, corresponding to vectors d_i on the unit $(n-1)$-sphere $\mathbb{S}^{n-1}$ centered at the origin.

The search for the optimal solution is a difficult task due both to the combinatorial nature of the problem and to the strong non-convexity given by the ℓ_0 conditions. We tackle this problem adopting the well established alternating optimization scheme [18], which consists in repeatedly executing the two steps:

- Step 1. Solve problem (1) for X only (fixing the dictionary D from previous step)
- Step 2. Solve problem (1) for D only (fixing X from previous step).

In particular, for sparse decomposition in Step 1 we use OMP because of its simplicity yet efficiency. Clearly, other sparse recovery methods could be adopted (e.g. MP, BP, Lasso). Experimentally, we observe that this choice does not affect R-SVD performances.

[1]Named after the ancient Greek myth of Damastes (known as Procrustes, the "stretcher"), son of Poseidon, who used to offer hospitality to the victims of his brigandage compelling them to fit into an iron bed by stretching or cutting off their legs.

Step 2 represents the core of the R-SVD method as detailed as follows.

A. Dictionary learning by Procrustes analysis

Let us first recall the idea of the Procrustes analysis. It consists in applying affine transformations (e.g., moving, stretching and rotating) to a given geometrical object in order to best fit the shape of another one. When the admissible transformations are restricted to orthogonal ones, it is referred to as Orthogonal Procrustes analysis [21].

In R-SVD method, after splitting the dictionary D into sub-dictionaries, the Orthogonal Procrustes analysis is applied to each of them to find the rotation (either proper or improper) that minimizes the least squares error. Consequently, each sub-dictionary is updated by the optimal affine transformation thus obtained.

Specifically, let $I \subset [m] := \{1, ..., m\}$ denote a subset of column or row indices. Given any index set I with size $s = |I|$, let $D_I \in \mathbb{R}^{n\times s}$ be the sub-dictionary of D formed by the *columns* indexed by I, and $X_I \in \mathbb{R}^{s\times L}$ the sub-matrix of X formed by the *rows* indexed by I; hence s is the size of sub-dictionary D_I. In this setting, we can decompose the product $DX = D_I X_I + D_{I^c} X_{I^c}$, with $I^c = [m] \setminus I$. By isolating the term $D_I X_I$ in $\|Y - DX\|_F^2$ and setting $E := Y - D_{I^c} X_{I^c}$, the optimization over the sub-dictionary $D_I \subset D$ corresponds to solving the problem

$$\underset{S\in\mathbb{R}^{n\times s}}{\operatorname{argmin}} \|E - SX_I\|_F^2 \qquad \text{subject to} \qquad S \subset \mathbb{S}^{n-1} \tag{2}$$

Our method aims at yielding a new sub-dictionary D'_I through an orthogonal transformation matrix R (i.e. $R^T R = I$) applied on D_I, namely $D'_I = RD_I$. The set of orthogonal matrices R of order n, called orthogonal group $O(n)$, can be partitioned into the special orthogonal subgroup $SO(n)$ formed by proper rotations, i.e. those with $\det R = 1$, and the set $O(n) \setminus SO(n)$ of improper rotations (or rotoreflections), i.e. those with $\det R = -1$. This leads to the minimization problem

$$\min_{R\in O(n)} \|E - RH\|_F^2 \tag{3}$$

where $H := D_I X_I \in \mathbb{R}^{n\times L}$. Notice that in denoting E and H we omit the dependence on I. The problem (3) is known as the *Orthogonal Procrustes problem* [21] and can be interpreted as finding the rotation of a subspace matrix H^T to closely approximate a subspace matrix E^T [25, §12.4.1]. The orthogonal Procrustes problem admits an optimal solution $\hat{R}$ which is the transposed orthogonal factor of the polar decomposition $EH^T = QP$, and can be computed as $\hat{R} = Q^T = VU^T$ from the singular value decomposition $EH^T = U\Sigma V^T \in \mathbb{R}^{n\times n}$.

Hence the rotation matrix we seek is $\hat{R} = VU^T$, the new dictionary D' has the old columns of D in the positions I^c and the new submatrix $D'_I = \hat{R}D_I$ in the positions I, while the new value of reconstruction error is $\|Y - D'X\|_F^2 = \|Y - D_{I^c}X_{I^c} - VU^T D_I X_I\|_F^2$.

At this point the method is quite straight-forward: at each dictionary update iteration (Step 2) partition the set of column indices $[m] = I_1 \sqcup I_2 \sqcup ... \sqcup I_G$ into G subsets, then split D accordingly into sub-dictionaries D_{I_g}, $g = 1, ..., G$, and update

every sub-dictionary. These updates can be carried out either in parallel or sequentially with some order. We have chosen the sequential update with ascending order of atom popularity, i.e. sorting the indices $i \in [m]$ w.r.t. the usage of atom d_i, computable as ℓ_0-norm of the i-th row in X. Other sorting criteria could be taken (e.g. random, cumulative coherence, clustering by absolute cosine similarity), even though we experimentally observed that this choice is not relevant. For sake of simplicity we set uniformly the sub-dictionary size to $s = |I_g|$ for all g, possibly except the last sub-dictionary ($G = \lceil m/s \rceil$) if m is not a multiple of s: $|I_G| = m - Gs$. After processing all G sub-dictionaries, the method moves to the next iteration, and goes on until a stop condition is reached (eg. maximum number of iterations, empirical convergence criterion). The main algorithmic steps can be summarized as follows[2]:

1: Initialize dictionary D picking m examples from Y at random
2: **repeat**
3: Approximate sparse coding: $X = \text{argmin}_X \|Y - DX\|_F^2$ subject to $\|x_i\|_0 \leq k$ for $i = 1, ..., L$
4: Partition indices $[m] = I_1 \sqcup I_2 \sqcup ... \sqcup I_G$ sorting by atom popularity
5: **for** $g = 1, ..., G$ **do**
6: $J = I_g$; $E = Y - D_{J^c} X_{J^c}$; $H = D_J X_J$
7: $R = \text{argmin}_{R \in O(n)} \|E - RH\|_F^2$ by rank-s SVD
8: $D_J = RD_J$
9: **end for**
10: **until** stop condition

B. Computational speedup

A useful computational speedup in the update of every sub-dictionary D_I can be described as follows. Let us pre-compute $XY^T \in \mathbb{R}^{m \times n}$ and $XX^T \in \mathbb{R}^{m \times m}$ at the beginning of each dictionary update (Step 2). The matrix EH^T undergoing the SVD can be computed as $EH^T = D_I[X_I Y^T - X_I (X_{I^c})^T (D_{I^c})^T]$, where D_I and D_{I^c} come from previous update step, and the term $X_I Y^T$ is the submatrix formed by rows I of XY^T, while $X_I (X^T)_{I^c}$ by rows I and columns I^c of XX^T. With elementary matrix products this computation requires $O(sn(n+m-s))$ flops, that is preferred rather than $O(nmL)$ since $s < n < m \ll L$.

Notice that, since $\text{rank}\, HE^T \leq \text{rank}\, H \leq s$, it is not necessary to obtain the full SVD of HE^T, but rather truncate the decomposition to the first s singular vectors u_i, v_i: $\hat{R} = VU^T = \sum_{i=1}^{s} v_i u_i^T$. We thus use the truncated SVD algorithm by [26] based on structured random matrix that requires $O(n^2 \log s)$ flops. The computation of HE^T and its SVD is repeated $G = \lceil m/s \rceil$ times. The computational time of R-SVD is dominated by the pre-computation of XY^T and XX^T, and therefore taking into account the sparsity of matrix X the asymptotic estimate for one iteration of R-SVD is $T_{\text{R-SVD}}(k, n, m, L, s) = O((\frac{k^2}{s} + \frac{k}{s} n)mL)$, compared to K-SVD's iteration $T_{\text{K-SVD}}(k, n, m, L) = O((k^2 + n)mL)$.

[2]The Matlab code implementing the algorithm is available on the website http://phuselab.di.unimi.it/resources.php

III. Experimental analysis

In this section we test the proposed R-SVD algorithm devoting at first an in-depth analysis to how choosing the sub-dictionary size s defined above. Then we apply the method both on synthetic data, as in [24], and to ECG compression and image reconstruction tasks.

In the experiments with synthetic data, the dictionary $D \in \mathbb{R}^{n \times m}$ is randomly drawn, with i.i.d. standard Gaussian distributed entries and each column normalized to unit ℓ_2-norm. The training set $Y \in \mathbb{R}^{n \times L}$ is generated column-wise by L linear combinations of k dictionary atoms selected at random and by adding white Gaussian noise matrix N with various signal-to-noise ratio (SNR), i.e. $Y = DX + N$. We measure the performances of the algorithm in terms of the reconstruction error expressed as $\text{E}_{\text{SNR}} = 20 \log_{10}(\|Y\|_F / \|Y - \tilde{D}\tilde{X}\|_F)$ dB, where $\tilde{D}$ and $\tilde{X}$ are the learnt dictionary and the sparse encoding matrix respectively.

A. An insight on the sub-dictionary size s

It is naturally expected that the sub-dictionary size s affects both reconstruction quality and running time. In order to give some insight on this parameter, we run R-SVD algorithm on synthetic training sets by setting $L = 8000$, $D \in \mathbb{R}^{50 \times 100}$, $k = 5$ and SNR $= 30$ dB for noise N and letting s range in the interval $1 \div 25$. Notice that when $s = 1$, our method is similar to K-SVD [24] except in the recovery of the sparse coefficients yielded by SVD decomposition.

In Fig. 1 we report the reconstruction error E_{SNR} (solid curve), and the computational times of both R-SVD (dashed curve) and K-SVD (dotted line), all averaged over 100 trials. It can be noticed that R-SVD method behaves better near the value $s = 10$. We thus choose this tradeoff setting for the experimental assessments of the method in the following sections.

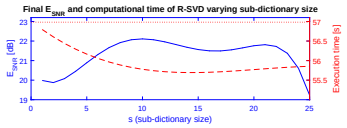

Fig. 1. R-SVD's dependency on the sub-dictionary size parameter s. Other experiment parameters are: training size $L = 8000$, dictionary size 50×100, additive noise of SNR $= 30$ dB, number of iterations $T = 200$. The lines (connecting points, for sake of readability) represent: average final E_{SNR} of the reconstructed dictionary (*solid blue curve*) w.r.t. the generating dictionary, computational time of the R-SVD (*dashed red curve*) and the K-SVD (*dotted red line*) in the dictionary learning task.

B. Comparative results on synthetic data

In order to test the R-SVD method and compare it to the K-SVD algorithm, we run experiments on random training instances. We consider dictionaries of size 50×100 and 100×200, dataset of size up to $L = 10000$ and sparsity $k = \{5, 10\}$. The algorithms K-SVD and R-SVD are run for $T = 200$ dictionary update iterations, that turns out to be sufficient to achieve empirical convergence of the performance measure. For each experimental setting we report the average error over 100 trials.

In Fig. 2 we highlight the learning trends of the two methods, plotting at each iteration count the $\mathrm{E_{SNR}}$ values on synthetic vectors $Y = DX + N$, varying the additive noise $\mathrm{SNR} = 10, 30, 50, \infty$ (no noise) dB. It can be seen that, after an initial transient, the gap between R-SVD and K-SVD increases with the iteration count, establishing a final gap of 2 dB or more in conditions of middle-low noise power ($\mathrm{SNR} \geq 30$ dB).

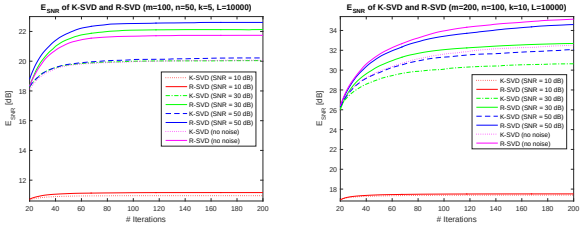

Fig. 2. Average reconstruction error $\mathrm{E_{SNR}}$ in sparse representation using dictionary learnt by K-SVD (non-solid lines) and R-SVD (solid lines), for $L = 10000$ synthetic vectors varying the additive noise power (in the legend). Averages are calculated over 100 trials and plotted versus update iteration count. *Left*: $D \in \mathbb{R}^{50\times 100}$ with sparsity $k = 5$, *Right*: $D \in \mathbb{R}^{100\times 200}$ with sparsity $k = 10$.

In order to explore the behaviour of R-SVD and K-SVD in a fairly wide range of parameter values, we report in Fig. 3 the gaps between their final ($T = 200$) reconstruction error $\mathrm{E_{SNR}}$, varying L in $2000 \div 10000$, noise SNR in $0 \div 60$ dB, and in case of no noise. Dictionary sizes, sparsity and number of trials are set as above. When the additive noise power is very

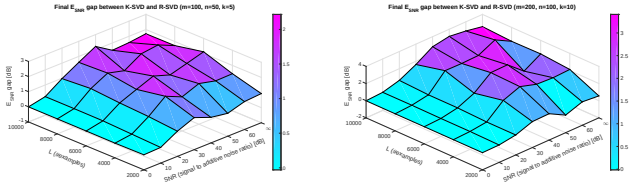

Fig. 3. Gap between final ($T = 200$) $\mathrm{E_{SNR}}$ of K-SVD and R-SVD obtained with all parameter combinations $L = 2000, 4000, 6000, 8000, 10000$ and $\mathrm{SNR} = 0, 10, 20, 30, 40, 50, 60, \infty$ (no noise). Results are averages over 100 trials; points are interpolated with coloured piece-wise planar surface for sake of readability. *Left*: $D \in \mathbb{R}^{50\times 100}$ with sparsity $k = 5$. *Right*: $D \in \mathbb{R}^{100\times 200}$ with sparsity $k = 10$.

high (eg. $\mathrm{SNR} = 0, 10$ dB) the two methods are practically comparable, probably because the presence of significant noise would mislead any learning algorithm. On the other hand, when the noise is quite low the R-SVD algorithm outperforms K-SVD with a gap up to 3 dB.

Moreover, it is useful to evaluate the number of correctly identified atoms in order to measure the ability of the learning algorithms in recovering the original dictionary D from the noise-affected data Y. This is accomplished by maximizing the matching between atoms d_i of the original dictionary with atoms $\tilde{d}_j \in \tilde{D}$ of the dictionary yielded by the algorithm: two atoms $(d_i, \tilde{d}_j)$ are considered matched when their cosine distance is small [24], i.e. precisely

$$1 - |d_i^T \tilde{d}_j| < \delta := 0.01.$$

In Table I we report the average number of recovered atoms on random instances. Notice that R-SVD performs slightly better independently of the additive noise power.

Number of recovered atoms

	SNR = 10		SNR = 30		SNR = 50		no nois
$n \times m$	K-SVD	R-SVD	K-SVD	R-SVD	K-SVD	R-SVD	K-SVD
50×100	94.52	97.37	92.15	94.08	92.1	93.84	92.07
100×200	195.82	199.02	192.42	194.98	192.49	194.57	192.87

TABLE I
AVERAGE NUMBER OF ATOMS CORRECTLY RECOVERED (MATCHED) BY K-SVD AND R-SVD ALGORITHMS AT VARIOUS SNR LEVELS OF ADDITIVE NOISE ON DICTIONARY D OF SIZE 50×100 AND 100×200. $L = 10000$, AND REMAINING PARAMETER VALUES AS IN FIG. 3.

C. Application to ECG compression

Recently, sparsity techniques have been applied to the compression of electrocardiogram (ECG) signals [27]. To highlight the benefit of the dictionary learning approach in this task, we tested the two methods K-SVD and R-SVD recasting the compression as a problem of sparse approximation with a dictionary.

In this experiment, the compression process is broken down into three stages. The former is a preprocessing step consisting in R-peak detection of the signal, and its normalization (i.e., filtering, zero-padding and centering) so as to split it into n-length normalized RR-segments. The second stage focuses on the dictionary learning where either R-SVD or K-SVD are used to train a dictionary on a group of RR-segments taken from an initial transient of the signal (train chunk). The latter stage concerns the encoding via sparse reconstruction of all RR-segments belonging to a broader interval of the signal (test chunk). This step is carried out referring to the learnt dictionaries, and applying the OMP algorithm. Naturally, in order to make the coding step easier, magnitudes and positions of non-null sparse coefficients are handled separately.

The compression level achieved by each method is measured as compression rate (CR), that is the ratio between the number of bits of the original signal and that of the compressed representation. Assuming that the test chunk y is composed of N q-bit resolution samples forming M RR-segments, we have $\mathrm{CR} = qN / \left(k\hat{q}M + M \log_2 \binom{m}{k}\right)$, where k is the sparsity level, $\hat{q}$ denotes the bit resolution of the quantized coefficients, m is the number of the atoms in the dictionary, and $\log_2 \binom{m}{k}$ is the binary coding length of coefficient positions. Being $\hat{y}$ the reconstructed version of y, the overall reconstruction quality measure is here specialized for ECG signal as $\mathrm{E_{SNR}} = 20 \log_{10} \frac{\|y\|_2}{\|\hat{y}-y\|_2}$ dB [3].

We conducted the experiments on two ECG records taken from the Long-Term ST Database in PhysioNet [28]. They are 24 hours long recordings taken from different subjects with the aid of Holter portable devices, all sampled at $f_s = 250$ Hz and $q = 12$-bit resolution. Performances are reported in Fig. 4. Upper plots show the $\mathrm{E_{SNR}}$ vs CR obtained from OMP, referring to dictionaries learnt by R-SVD or K-SVD, and to

[3]This technical change is due to the necessity of measuring the quality of signals formed by variable length segments.

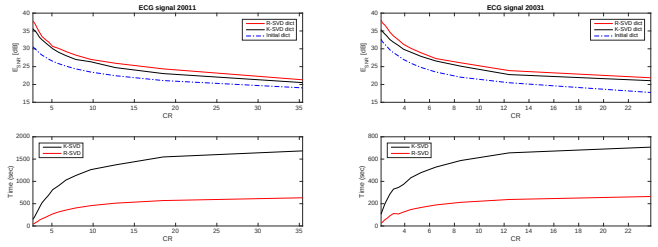

Fig. 4. Experiments on the ECG recordings s20011 and s20031 taken from the Long-Term ST Database. (Upper plots) $\mathrm{E_{SNR}}$ vs CR achieved by the sparsity-based OMP compressor on dictionary learnt by R-SVD, K-SVD or on a random untrained dictionary. (Lower plots) Computational time spent by the two techniques in the learning stage.

an untrained dictionary (i.e. randomly picking m RR-segments from the train chunk). Lower plots report the computational time spent by the two techniques in the learning stage. To make the experiment realistic, we set $n = f_s$, $m = 5n$ and $\hat{q} = q$, the dictionary training is carried out on $L = 5000$ RR-segments (about 120 minutes), while the test chunk is composed of $M = 15000$ RR-segments (about 4 hours). In order to analyze a wide range of sparsity levels, we varied k in the interval $5 \div 80$ with stepsize 1.

Such experiments, besides confirming that trained dictionaries behave better than untrained one, prove the effectiveness of the R-SVD training method that outperforms K-SVD both in training ability and learning time.

D. Application to image modeling

Sparse representation of images via learnt overcomplete dictionaries is a well-known topic in the fields of image processing and computer vision [29]. In particular, in problems such as image denoising or compression [2], [30], we are interested in constructing efficient representations of patches (i.e. small portions of images) as a combination of as few as possible typical patterns (atoms) learnt from the data itself. Naturally, the referred dictionary is crucial for the effectiveness of the method. Here, we compare performances achieved referring to the patch dictionaries learnt by either the well-known K-SVD or the R-SVD methods.

Given a pool of image patches P, a pre-processing is first applied aiming at both removing the patch mean intensity and reshaping all the patches to vectors in $\mathbb{R}^n$.

The learning phase is carried out by setting the number of iterations to 50, and varying the sparsity k in $5 \div 30$. The initial dictionaries for the two algorithms have size $n \times 1.5n$, while training and test sets have size $L = 2n$ and $M = 4n$ respectively; they are all made up of randomly selected patches from P.

Experimentally, we considered patches of size 9×9 ($n = 81$) and 16×16 ($n = 256$). In both cases we randomly extracted $100,000$ patches from 500 images of Caltech 101 [31] and Berkeley segmentation image database [32]. In Fig. 5 we plot the $\mathrm{E_{SNR}}$ trends of the learning phase on the training sets. We can observe that, since the first iterations, R-SVD behaves better than K-SVD, maintaining a positive gap at convergence, especially in the cases of lower sparsity.

In Table II we report the reconstruction quality in terms of $\mathrm{E_{SNR}}$ attained in the test phase using the dictionaries learnt by K-SVD and R-SVD, and applying OMP. All results in the two phases are averaged on 50 trials. As we can observe, the R-SVD method systematically exceeds K-SVD referring to both patch sizes and to several sparsity degrees. These results further confirm the effectiveness and generality of the R-SVD method.

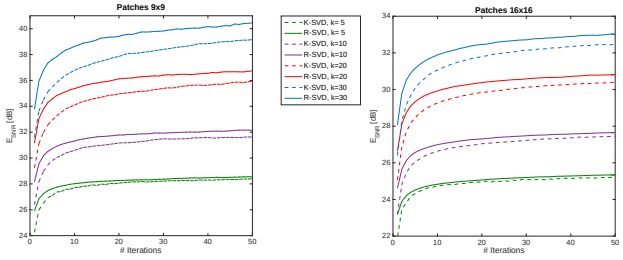

Fig. 5. Average reconstruction error $\mathrm{E_{SNR}}$ for patches 9×9 and 16×16 using dictionary learnt by K-SVD (dashed lines) and R-SVD (solid lines). Averages are calculated over 50 trials and plotted versus update iteration count. Considered sparsity levels: $k = \{5, 10, 20, 30\}$.

Reconstruction error $\mathrm{E_{SNR}}$

n	k	D_{init}	$D_{\text{K-SVD}}$	$D_{\text{R-SVD}}$
81	5	16.76	19.74	20.16
81	10	18.34	21.94	22.22
81	20	20.78	24.73	25.12
81	30	22.88	27.38	27.65
256	5	14.84	17.40	17.74
256	10	15.91	18.96	19.27
256	20	17.49	20.96	21.21
256	30	18.45	22.02	22.34

TABLE II

AVERAGE $\mathrm{E_{SNR}}$ OBTAINED ON THE IMAGE TEST SETS. n: LINEAR PATCH DIMENSION. k: SPARSITY LEVEL. LAST THREE COLUMNS ARE $\mathrm{E_{SNR}}$ ACHIEVED WITH INITIAL DICTIONARY (D_{INIT}), DICTIONARY LEARNT BY THE K-SVD METHOD ($D_{\text{K-SVD}}$) AND DICTIONARY LEARNT BY THE R-SVD METHOD ($D_{\text{R-SVD}}$).

IV. CONCLUSIONS

In this paper we propose a new technique, namely R-SVD, of dictionary learning for sparse coding. It preserves the well established iterative alternating scheme adopted for example in the K-SVD algorithm: one step is for the coding of sparse coefficients, and one for the dictionary optimization promoting sparsity. The main novelty of R-SVD concerns how it tackles the dictionary optimization step: instead of choosing single best atoms via SVD, it transforms groups of atoms (i.e., sub-dictionaries) through the best rotations found in the spirit of the Orthogonal Procrustes analysis, in order to minimize the representation error.

This approach shows its effectiveness in experiments both in case of synthetic and natural data. In particular, for the latter case, we consider the signal and image processing domain, showing good performances on the tasks of ECG compression and image modeling. We also discuss, through empirical tests, how to choose the sub-dictionary size in order to improve both sparse solution quality and computational costs.

Some open issues remain to be studied. For instance, how to tackle problem (2), i.e. find the best sub-dictionaries adopting more general transformations other than rotations with the Procrustes analysis approach. Another question concerns the resolution of problem (3) possibly through approximation techniques guaranteeing a better computational efficiency.

REFERENCES

[1] A. Adamo, G. Grossi, R. Lanzarotti, and J. Lin, "ECG compression retaining the best natural basis k-coefficients via sparse decomposition," *Biomed. Signal Process. Control*, vol. 15, pp. 11–17, 2015.

[2] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *2006 IEEE Comp. Soc. Conf. CVPR*, vol. 1, 2006, pp. 895–900.

[3] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Adv. Neural. Inf. Process. Syst.*, 2006, pp. 609–616.

[4] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constr. Approx.*, vol. 13, no. 1, pp. 57–98, 1997.

[5] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, pp. 3397–3415, 1993.

[6] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Systems and Computers*. IEEE Comput. Soc. Press, 1993, pp. 40–44.

[7] 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.

[8] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Stat. Soc. Series B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

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

[10] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.

[11] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part i: theory," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 539–554, 2006.

[12] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, 2005.

[13] G. Easley, D. Labate, and W.-Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Appl. Comput. Harmon. Anal.*, vol. 25, no. 1, pp. 25–46, 2008.

[14] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.

[15] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Res.*, vol. 37, no. 23, pp. 3311 – 3325, 1997.

[16] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

[17] K. Engan, S. O. Aase, and J. H. Husøy, "Method of optimal directions for frame design," in *Proc. 1999 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 5, 1999, pp. 2443–2446.

[18] ——, "Designing frames for matching pursuit algorithms," in *Proc. 1998 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 3, 1998, pp. 1817–1820.

[19] K. Engan, K. Skretting, and J. H. Husøy, "Family of iterative ls-based dictionary learning algorithms, ils-dla, for sparse signal representation," *Digital Signal Process.*, vol. 17, no. 1, pp. 32–49, 2007.

[20] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2121–2130, 2010.

[21] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*. Oxford University Press, 2004, vol. 3.

[22] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[23] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallogr., Sect. A: Found. Crystallogr.*, vol. 32, no. 5, pp. 922–923, 1976.

[24] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Process.*, vol. 54, pp. 4311–4322, 2006.

[25] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. John Hopkins University Press, 1996.

[26] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[27] G. Grossi, R. Lanzarotti, and J. Lin, "High-rate compression of ecg signals by an accuracy-driven sparsity model relying on natural basis," *Digit. Signal Process.*, vol. 45, no. C, pp. 96–106, 2015.

[28] G. Moody and R. Mark, "The MIT-BIH Arrhythmia Database on CD-ROM and software for use with it," in *Proc. Computers in Cardiology 1990*, September 1990, pp. 185–188.

[29] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition." *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[30] M. Xu, J. Lu, and W. Zhu, "Sparse representation of texture patches for low bit-rate image compression," in *2012 IEEE Visual Communications and Image Processing (VCIP)*, 2012, pp. 1–6.

[31] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Comput. Vision Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[32] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 2001, pp. 416–423.