EFFICIENT LEARNING OF DICTIONARIES WITH LOW-RANK ATOMS

Saiprasad Ravishankar, Brian E. Moore, Raj Rao Nadakuditi, and Jeffrey A. Fessler

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

ABSTRACT

Sparsity-based techniques have been popular in many applications in signal and image processing. In particular, the data-driven adaptation of sparse signal models such as the synthesis model has shown promise in applications. However, dictionary learning problems are typically nonconvex and NP-hard, and the usual alternating minimization approaches for learning are often expensive and lack convergence guarantees. In this work, we investigate efficient methods for learning *structured* synthesis dictionaries. In particular, we model the atoms (columns) of the dictionary, after reshaping, as low-rank. We propose a block coordinate descent algorithm for our dictionary learning model that involves efficient optimal updates. We also provide a convergence analysis of the proposed method for a highly nonconvex problem. Our numerical experiments show the usefulness of our schemes in inverse problem settings, such as in dynamic MRI and inpainting.

***Index Terms*—** Dictionary learning, structured models, sparse representations, convergence analysis, inverse problems.

1. INTRODUCTION

The sparsity of signals and images in a transform domain or dictionary has been extensively exploited in applications such as compression, denoising, and inverse problems in imaging and image processing. In particular, the data-driven adaptation of sparse signal models such as the synthesis model has shown promise in numerous applications [1–4]. Given a set of signals (or vectorized image patches) $\{y_i\}_{i=1}^N$ that are represented as columns of a training matrix $Y \in \mathbb{C}^{n\times N}$, the goal of dictionary learning (DL) is to learn a dictionary $D \in \mathbb{C}^{n\times J}$ and a matrix $X \in \mathbb{C}^{J\times N}$ of sparse codes such that $Y \approx DX$. The DL problem is often formulated as follows [5]:

$$\text{(P0)}\ \min_{D,X}\ \|Y - DX\|_F^2\ \text{ s.t. }\ \|x_i\|_0 \leq s\ \forall i,\ \|d_j\|_2 = 1\,\forall j,$$

where x_i and d_j denote the ith column of X and the jth column (or atom) of D respectively, and s denotes a target sparsity level for each signal. The ℓ_0 "norm" measures sparsity and counts the number of non-zero entries in a vector. The columns of D are set to unit norm to avoid the scaling ambiguity [6]. Various alternative versions of (P0) exist that replace the ℓ_0 "norm" with other sparsity-promoting functions, or enforce additional properties on the dictionary [7–9], or enable dictionary learning in an online manner [10].

Dictionary learning algorithms [5, 10–14] typically attempt to solve (P0) or its variants in an alternating manner by performing a sparse coding step (updating X) followed by a dictionary update step (updating D). Some algorithms also partially update the coefficients in X in the dictionary update step, while a few recent methods attempt to solve for the variables jointly and iteratively [15]. However, (P0) is non-convex and NP-hard, and most popular algorithms such as K-SVD [5] lack proven convergence guarantees, and tend to be computationally expensive. Some recent works [16–19] have studied the convergence of specific DL algorithms (typically making restrictive assumptions such as noiseless data, etc., for their convergence results), but these approaches have not been demonstrated to be advantageous in applications such as inverse problems. Bao et al. [18] find that their method, although a fast proximal scheme, denoises less effectively than K-SVD.

In this work, we propose a novel framework for structured dictionary learning. We model the atoms of the dictionary, after reshaping them into matrices, as low-rank. We also use an ℓ_0 sparsity penalty for the coefficients. Although the proposed DL formulation is highly nonconvex, we develop an efficient block coordinate descent algorithm for it and present a convergence analysis for the approach. Our numerical experiments demonstrate the suitability and usefulness of learning low-rank atom dictionaries in applications (inverse problems) involving limited data.

2. PROBLEM FORMULATION AND ALGORITHM

This section presents our DL problem formulation with structured (low-rank) atoms and an efficient algorithm for it.

2.1. Dictionary Learning Problem Formulation

We consider a dictionary learning formulation with a sparsity penalty in this work. In particular, we define $C \triangleq X^H$ in (P0), and replace the ℓ_0 "norm" constraints with an *overall* sparsity penalty $\|X\|_0 \triangleq \sum_{i=1}^N \|x_i\|_0 = \|C\|_0 = \sum_{j=1}^J \|c_j\|_0$. In addition, we consider a form of structured dictionary learning for image or video patches, wherein the columns $d_j \in \mathbb{C}^n$ of D, after being reshaped into matrices, are low-rank. We refer to this as the DIctioNary with lOw-ranK AToms (DINO-KAT) model. When the training matrix Y consists of vectorized versions of $n_1 \times n_2$ (with $n = n_1 n_2$) image patches, the dictionary atom vectors are reshaped (by stacking column-wise the vector entries) into similarly sized matrices. Spatiotemporal (3D) patches of videos typically have correlations along the time axis, and so may be well represented by a dictionary with low-rank space-time (reshaped) atoms. Denoting by $R(\cdot)$ the operator that reshapes an atom into a matrix, our problem formulation for DL is as follows:

$$\begin{aligned}\text{(P1)}\ &\min_{D,C}\ \left\|Y - DC^H\right\|_F^2 + \lambda^2 \|C\|_0\\ &\quad \text{s.t. } \operatorname{rank}(R(d_j)) \leq r,\ \|d_j\|_2 = 1,\ \|c_j\|_\infty \leq L\ \forall j. \quad (1)\end{aligned}$$

Here, λ^2 with $\lambda > 0$, is a sparsity regularization parameter and $r > 0$ denotes the maximum allowed rank for reshaped atoms.

The objective in (P1) is invariant to joint scaling of any pair (d_j, c_j) as $(\alpha d_j, \alpha^{-1} c_j)$, for $\alpha \neq 0$. Therefore, similar to Problem (P0), the constraint $\|d_j\|_2 = 1$ helps remove this scaling ambiguity.

This work was supported in part by the following grants: ONR grant N00014-15-1-2141, DARPA Young Faculty Award D14AP00086, ARO MURI grants W911NF-11-1-0391 and 2015-05174-05, NIH grant U01 EB01875301, and a UM-SJTU seed grant.

978-1-5090-4545-7/16/$31.00 ©2016 IEEE

The ℓ_∞ constraints in (P1) prevent pathologies that could theoretically arise (e.g., unbounded algorithm iterates) due to the objective being non-coercive [20]. In practice, we set L very large, and the constraint is typically inactive.

Unlike the sparsity constraints in (P0), Problem (P1) penalizes the number of non-zeros in the (entire) coefficient matrix, allowing variable sparsity levels across the training signals. For example, in imaging or image processing applications, the dictionary is usually learned on (overlapping) image patches. Patches from different regions of an image typically contain different amounts of information, and thus enforcing a fixed or common sparsity for various patches (as in (P0)) does not reflect typical image properties (i.e., is restrictive) and usually leads to poor performance in applications.

When $r = \min(n_1, n_2)$ for $R(d_j) \in \mathbb{C}^{n_1 \times n_2}$, the rank constraints in (P1) are inactive, and Problem (P1) corresponds to an unstructured DL formulation [20]. Structured DINO-KAT models (i.e., with small rank r) learned using (P1) may be less prone to over-fitting problems in applications involving limited or corrupted data. We illustrate this through some applications in Section 3.

2.2. Algorithm and Computational Cost

We propose an iterative block coordinate descent method [21] for (P1) that updates the coefficient columns c_j (of C) and atoms d_j (of D) sequentially. Specifically, for each $1 \leq j \leq J$, we first solve (P1) with respect to c_j, keeping the other variables fixed (*sparse coding step*). Once c_j is updated, we solve (P1) with respect to d_j, keeping all other variables fixed (*dictionary atom update step*).

2.2.1. Sparse Coding Step

Here, we minimize (P1) with respect to c_j. This leads to the following problem, where the matrix $E_j \triangleq Y - \sum_{k \neq j} d_k c_k^H$ is computed using the most recent estimates of other atoms and coefficients:

$$\min_{c_j} \; \left\| E_j - d_j c_j^H \right\|_F^2 + \lambda^2 \left\| c_j \right\|_0 \;\; \text{s.t.} \;\; \left\| c_j \right\|_\infty \leq L. \tag{2}$$

The solution to the above sparse coding problem is stated in the following proposition. Its proof is identical to that for Proposition 1 in [20]. The thresholding operator $H_\lambda(\cdot)$ is defined as follows, where $b \in \mathbb{C}^N$ and the subscript i indexes vector entries:

$$\left(H_\lambda(b)\right)_i = \begin{cases} 0, & |b_i| < \lambda \\ b_i, & |b_i| \geq \lambda \end{cases} \tag{3}$$

We choose $L > \lambda$ here and let 1_N denote a vector of ones of length N. The operation "$\odot$" denotes element-wise multiplication, and $z = \min(a, u)$ denotes element-wise minimum. For a vector $c \in \mathbb{C}^N$, $e^{j\angle c} \in \mathbb{C}^N$ is computed element-wise, with "$\angle$" denoting the phase.

Proposition 1 *Suppose $L > \lambda$, and given $E_j \in \mathbb{C}^{n \times N}$ and $d_j \in \mathbb{C}^n$, a global minimizer of Problem* (2) *is*

$$\hat{c}_j = \min\left(\left|H_\lambda\left(E_j^H d_j\right)\right|, L 1_N\right) \odot e^{j\angle E_j^H d_j}. \tag{4}$$

This solution is unique if and only if the vector $E_j^H d_j$ has no entry with magnitude exactly equal to λ.

2.2.2. Dictionary Atom Update Step

In this step, we optimize (P1) with respect to the atom d_j, holding other variables fixed, leading to the following problem:

$$\min_{d_j} \; \left\| E_j - d_j c_j^H \right\|_F^2 \;\; \text{s.t.} \;\; \text{rank}\left(R(d_j)\right) \leq r, \; \left\| d_j \right\|_2 = 1. \tag{5}$$

The following Proposition 2 provides the solution to Problem (5). It relies on the full singular value decomposition (SVD) of an appropriate matrix. We assume $R(d_j) \in \mathbb{C}^{n_1 \times n_2}$, and let σ_i denote the ith entry on the main diagonal of the matrix Σ.

Proposition 2 *Given $E_j \in \mathbb{C}^{n \times N}$ and $c_j \in \mathbb{C}^N$, let $U_r \Sigma_r V_r^H$ denote an optimal rank-r approximation to $R\left(E_j c_j\right) \in \mathbb{C}^{n_1 \times n_2}$ that is obtained using the r leading singular vectors and singular values of the full SVD $R\left(E_j c_j\right) \triangleq U \Sigma V^H$. Then, a global minimizer in Problem* (5)*, upon reshaping, is*

$$R(\hat{d}_j) = \begin{cases} \frac{U_r \Sigma_r V_r^H}{\left\| \Sigma_r \right\|_F}, & \text{if } c_j \neq 0 \\ W, & \text{if } c_j = 0 \end{cases} \tag{6}$$

where W is the reshaped first column of the $n \times n$ identity matrix. The solution is unique if and only if $c_j \neq 0$, and $\sigma_r > \sigma_{r+1}$ or $\sigma_r = 0$.

Proof: First, because $\left\| d_j \right\|_2 = 1$, the following result holds:

$$\left\| E_j - d_j c_j^H \right\|_F^2 = \left\| E_j \right\|_F^2 + \left\| c_j \right\|_2^2 - 2\,\text{Re}\left\{ d_j^H E_j c_j \right\} \tag{7}$$

Upon substituting (7) into (5), Problem (5) simplifies to

$$\max_{d_j \in \mathbb{C}^n} \text{Re}\left\{ \text{tr}\left(R(d_j)^H R\left(E_j c_j\right) \right) \right\} \;\; \text{s.t.} \; \text{rank}\left(R(d_j)\right) \leq r, \quad \left\| d_j \right\|_2 = 1. \tag{8}$$

Next, let $R(d_j) = G \Gamma B^H$, and $R\left(E_j c_j\right) = U \Sigma V^H$ be full SVDs, with γ_i and σ_i the entries on the main diagonals of Γ and Σ, respectively. The problem then becomes

$$\max_{\Gamma} \; \max_{G, B} \; \text{Re}\left\{ \text{tr}\left(B \Gamma^T G^H U \Sigma V^H \right) \right\} \quad \text{s.t.} \;\; \text{rank}(\Gamma) \leq r, \left\| \Gamma \right\|_F = 1, G^H G = I, B^H B = I. \tag{9}$$

For the inner maximization, we use $\text{Re}\left\{ \text{tr}\left(B \Gamma^T G^H U \Sigma V^H \right) \right\} \leq \text{tr}\left(\Gamma^T \Sigma \right)$ [22], with the upper bound attained with $G = U$ and $B = V$. The remaining problem with respect to Γ is then

$$\max_{\{\gamma_i\}} \sum_{i=1}^{r} \gamma_i \sigma_i \;\; \text{s.t.} \;\; \sum_{i=1}^{r} \gamma_i^2 = 1, \gamma_j = 0, \; \forall \, j > r. \tag{10}$$

Using the Cauchy Schwarz inequality, $\hat{\gamma}_i = \sigma_i / \sqrt{\sum_{i=1}^{r} \sigma_i^2}$ for $1 \leq i \leq r$, and $\hat{\gamma}_i = 0$ for $i > r$ is clearly optimal. The derived solution for the optimal $R(\hat{d}_j)$ then simply corresponds to a normalized version of the rank-r approximation to $R\left(E_j c_j\right)$. Clearly, the solution in (8) is unique if and only if $E_j c_j \neq 0$, and $\sigma_r > \sigma_{r+1}$ or $\sigma_r = \sigma_{r+1} = 0$. Any $d \in \mathbb{C}^n$ satisfying the constraints in (8) is a (non-unique) minimizer when $E_j c_j = 0$. In particular $R(\hat{d}_j) = W$ works.

Lastly, to complete the Proposition's proof, we show that $E_j c_j = 0$ in our algorithm if and only if $c_j = 0$. Since c_j here was obtained as a minimizer in the preceding sparse coding step (2), we have the following result $\forall \, c \in \mathbb{C}^N$ with $\left\| c \right\|_\infty \leq L$ and $\tilde{d}_j$ denoting the jth atom in the preceding sparse coding step:

$$\left\| E_j - \tilde{d}_j c_j^H \right\|_F^2 + \lambda^2 \left\| c_j \right\|_0 \leq \left\| E_j - \tilde{d}_j c^H \right\|_F^2 + \lambda^2 \left\| c \right\|_0. \tag{11}$$

If $E_j c_j = 0$, the left hand side above is $\left\| E_j \right\|_F^2 + \left\| c_j \right\|_2^2 + \lambda^2 \left\| c_j \right\|_0$, which is clearly minimal when $c_j = 0$. Thus, when $E_j c_j = 0$, we must have $c_j = 0$. ■

2.2.3. *Overall Algorithm and Computational Cost*

The overall block coordinate descent DINO-KAT algorithm involves J sparse coding and dictionary atom update steps in each outer iteration. Assuming $J \propto n$ and $N \gg J, n$, the cost per iteration of the algorithm scales as $O(Nn^2)$. This cost is dominated by various matrix-vector products. The costs of the truncated hard-thresholding (4) and low-rank approximation (6) steps are negligible. The per-iteration cost for our method is lower than that for learning an $n \times J$ dictionary D in (P0) using K-SVD [5,23], which scales (with $s \propto n$ and $J \propto n$) as $O(Nn^3)$. Our algorithms also converge quickly in practice and outperform K-SVD in applications [20].

2.3. Convergence of DINO-KAT Learning Algorithm

We briefly present results on the convergence behavior of the proposed algorithm. The proofs of the results in this section follow using similar arguments as in the proofs of related results in [20].

The constraints $\operatorname{rank}(R(d_j)) \leq r$, $\|d_j\|_2 = 1$, and $\|c_j\|_\infty \leq L$ in (P1) can instead be added as penalties in the cost by using barrier functions $\phi(d_j)$, $\chi(d_j)$, and $\psi(c_j)$, respectively, that take the value $+\infty$ when the corresponding constraint is violated, and are zero otherwise. Problem (P1) is then written in unconstrained form with objective

$$g(C,D) = g\left(c_1,...,c_J,d_1,...,d_J\right) = \left\|Y - DC^H\right\|_F^2 + \sum_{j=1}^{J} \left\{\lambda^2 \left\|c_j\right\|_0 + \phi(d_j) + \chi(d_j) + \psi(c_j)\right\} \qquad (12)$$

We have the following monotonicity and consistency result.

Theorem 1 *Let $\left\{C^t, D^t\right\}$ denote the iterate sequence generated by the algorithm with training data $Y \in \mathbb{C}^{n\times N}$ and initial (C^0, D^0). Then, the objective sequence $\left\{g^t\right\}$ with $g^t \triangleq g\left(C^t, D^t\right)$ is monotone decreasing and converges to a finite value, say $g^* = g^*(C^0, D^0)$. Moreover, the iterate sequence $\left\{C^t, D^t\right\}$ is bounded, and all its accumulation points are equivalent in the sense that they achieve the same objective value g^*.*

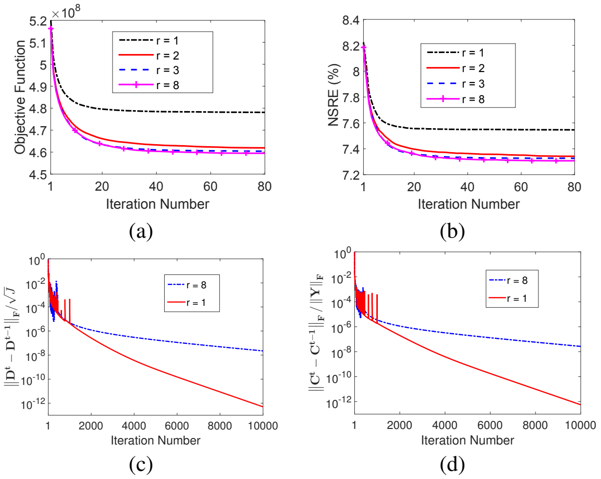
Theorem 1 establishes that for each initialization, all the accumulation points of the (bounded) iterate sequence of the algorithm achieve the same value g^* of the objective, and are equivalent. (g^* could vary with initalizations.) Because the distance between a bounded sequence and its compact set of accumulation points converges to zero, we have the following corollary.

Corollary 1 *For each (C^0, D^0), the iterate sequence in the algorithm converges to an equivalence class of accumulation points.*

Finally, the following theorem establishes that the iterates in our algorithm converge to the set of critical points [24] (or generalized stationary points) of $g(C, D)$. Here, σ_k denotes the kth singular value in the full SVD of a matrix.

Theorem 2 *Let $\left\{C^t, D^t\right\}$ denote the bounded iterate sequence in the algorithm with training data Y and initial (C^0, D^0). Suppose each accumulation point (C, D) of the iterate sequence is such that for each $1 \leq j \leq J$ with $E_j \triangleq Y - DC^H + d_j c_j^H$, the vector $E_j^H d_j$ has no entry with magnitude λ, and $\sigma_r\left(R\left(E_j c_j\right)\right) > \sigma_{r+1}\left(R\left(E_j c_j\right)\right)$ or $\sigma_r\left(R\left(E_j c_j\right)\right) = 0$. Then, every accumulation point of the iterate sequence is a critical point of $g(C, D)$. Moreover, the sequences with terms $\left\|D^t - D^{t-1}\right\|_F$ and $\left\|C^t - C^{t-1}\right\|_F$ respectively, both converge to zero.*

Fig. 1. Images: Barbara, Boat, Hill, and a Microscopy image [25].



(a)

(b)

(c)

(d)

Fig. 2. DL Algorithm behavior: (a) Objective function; (b) NSRE; (c) $\left\|D^t - D^{t-1}\right\|_F / \sqrt{J}$; and (d) $\left\|C^t - C^{t-1}\right\|_F / \left\|Y\right\|_F$.

Theorem 2 says that $\left\|D^t - D^{t-1}\right\|_F \to 0$ and $\left\|C^t - C^{t-1}\right\|_F \to 0$, which are necessary but not sufficient conditions for the convergence of the sequences $\left\{D^t\right\}$ and $\left\{C^t\right\}$. Although Theorem 2 assumes simple conditions (e.g., nondegenerate singular values) on the accumulation points, we conjecture that these conditions hold for each accumulation point with probability 1 when the training signals are drawn i.i.d. from an absolutely continuous probability measure.

3. NUMERICAL EXPERIMENTS

This section presents numerical results illustrating the convergence of the proposed DL method and its application to inverse problems.

3.1. Convergence Behavior

To study the practical convergence behavior of the proposed algorithm for (P1), we extracted 3×10^4 training patches of size 8×8 from randomly chosen locations in the images Barbara, Boat, and Hill shown in Fig. 1. We used (P1) with $\lambda = 69$ to learn a 64×256 dictionary for the data, with reshaped atoms of size 8×8. We set $C^0 = 0$, and D^0 to be the overcomplete DCT [1,26].

Fig. 2 shows the behavior of the algorithm for various choices of atom rank r. The objective in (P1) converged (Fig. 2(a)) monotonically and quickly over the iterations. Fig. 2(b) shows the normalized sparse representation error (NSRE) $\left\|Y - DC^H\right\|_F / \left\|Y\right\|_F$ for the data. (The sparsity $\left\|C\right\|_0 / Nn$ stayed at about 3% during the algorithm iterations for all choices of r.) The NSRE improved significantly beyond the first iteration, indicating the success of the proposed DL scheme. Importantly, the NSRE values achieved for small values of r (DINO-KAT cases) are very similar to the value in the full-rank ($r = 8$) case. This suggests that the low-rank model on reshaped dictionary atoms, despite being a constrained model, can effectively model properties of natural images. Lastly, both $\left\|D^t - D^{t-1}\right\|_F$ (Fig. 2(c)) and $\left\|C^t - C^{t-1}\right\|_F$ (Fig. 2(d)) converge towards 0, as predicted by Theorem 2, with quicker conver-

Case	Initial	Cubic	Fixed D	$r = 8$	$r = 3$	$r = 2$	$r = 1$
50%	11.1	36.9	34.8	**37.9**	**37.9**	**37.9**	**37.9**
30%	9.7	34.9	31.9	35.6	35.9	**36.0**	35.9
20%	9.1	33.4	30.2	34.6	**34.9**	34.8	34.8
10%	8.6	31.0	27.8	32.3	32.3	32.4	**32.6**

Table 1. Inpainting PSNR values in dB (at various percentages of measured pixels) for the initial image ($A^{\dagger}y$), the result with cubic interpolation, the results using (P2) with $r = 1$, $r = 2$, $r = 3$, and $r = 8$, and for the reconstruction obtained with fixed D (initialization) in our algorithm. Results are for the Microscopy image. The best PSNRs are marked in bold.

gence observed for the low-rank case as there are fewer degrees of freedom to learn.

3.2. Inverse Problem: Blind Compressed Sensing

In compressed sensing (CS) [27], the goal is to recover an image $x \in \mathbb{C}^p$ from its measurements $y = Ax + h$, where $A \in \mathbb{C}^{m \times p}$ with $m \ll p$ is a known sensing matrix, and h denotes noise. CS methods reconstruct the image (or video) by modeling it (or its patches) as sparse in a known transform or dictionary. Here, we consider blind compressed sensing (BCS) [28], where the sparse model is assumed unknown a priori. The image and the model are jointly estimated in BCS. We propose the following BCS problem based on (P1):

$$\text{(P2)} \quad \min_{x,D,B} \; \nu \left\| Ax - y \right\|_2^2 + \sum_{i=1}^{N} \left\| P_i x - D b_i \right\|_2^2 + \lambda^2 \left\| B \right\|_0$$
$$\text{s.t.} \quad \left\| b_i \right\|_\infty \leq L, \; \operatorname{rank}\left(R(d_j)\right) \leq r, \; \left\| d_j \right\|_2 = 1 \; \forall \, i, j.$$

Here, $P_i x$ is a (vectorized) patch of x, and B is a matrix with sparse codes b_i as its columns. We propose an algorithm for (P2) that alternates between updating (D, B) and x. In the first step, x is fixed, and the problem reduces to DL using (P1). The second step involves a simple least squares problem in x that can be solved either directly or using iterative solvers such as the proximal gradient method.

Here, we study the usefulness of (P2) for dynamic MRI (dMRI) and compressive scanning electron microscopy (SEM) [29].

A) Compressive SEM. We consider the SEM image [25] in Fig. 1 and simulate CS (inpainting) by sampling a subset of image pixels. We used (P2) with a 64×20 D learned on 8×8 overlapping image patches using 100 alternations between (D, B) and x with $\nu = 10^7$ and $\lambda = 0.05$. (We use larger λ values during initial alternations, which accelerates convergence.) We update (D, B) using 1 iteration of the algorithm for (P1). We set the initial $x = A^{\dagger}y$, the initial $B = 0$, and the initial D was a 64×20 DCT (generated as in [26]).

Table 1 shows the PSNR values at various undersampling factors for reconstructions obtained using our method, and with cubic interpolation (using Matlab's *griddata* function), and using the proposed method with fixed D (fixed to initialization). The proposed BCS scheme clearly achieves better reconstructions compared to cubic interpolation or conventional CS (fixed D). Importantly, in cases involving very limited data, enforcing the low-rank constraint ($r = 1, 2, 3$) on reshaped (8×8) dictionary atoms leads to better PSNRs compared to the unstructured ($r = 8$) case.

B) CS Dynamic MRI. We perform simulations with the multi-coil Cartesian-sampled cardiac perfusion data used in prior work [30]. Fully-sampled data with an image matrix size of 128×128 and 40 temporal frames were retrospectively undersampled (in k-t space) using a different variable-density random Cartesian undersampling pattern for each time frame. We use normalized root mean square error (NRMSE), defined as $\left\| x_{\text{recon}} - x_{\text{ref}} \right\|_2 / \left\| x_{\text{ref}} \right\|_2$, where x_{ref} is a reference reconstruction computed from the fully-sampled data and x_{recon} the reconstruction from the undersampled data, as our performance metric. We compare the performance of the proposed method to that of the recent L+S method [30, 31], where the dynamic data is modeled as a sum of a low-rank (L) and a sparse (S) (with respect to a temporal Fourier transform) component. For the L+S method, the parameters λ_L and λ_S were tuned to obtain good NRMSE in our experiments. For the proposed method for (P2), we use spatiotemporal patches of size $8 \times 8 \times 5$ with spatial and temporal patch overlap strides of 2 pixels, $\nu = 66.67$, $\lambda = 0.025$, and we initialize the algorithm by setting x to be the output of the L+S method, D to be the 320×320 DCT matrix, and $B = 0$.

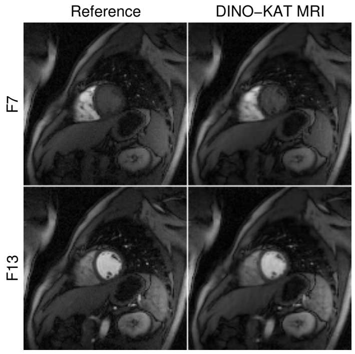


Fig. 3. 8x undersampling: Frames 7 and 13 of the proposed DINO-KAT MRI ($r = 1$) reconstruction along with the reference frames.

Acceleration	4x	8x	12x	16x	20x	24x
NRMSE (L+S) %	10.93	14.00	15.80	18.87	21.33	23.36
NRMSE (Fixed D) %	11.29	13.76	15.33	18.31	20.77	22.82
NRMSE (r = 5) %	10.85	13.08	14.37	17.01	19.19	21.35
NRMSE (r = 1) %	**10.57**	**12.90**	**14.20**	**16.77**	**18.74**	**20.91**
Gain over L + S (dB)	0.29	0.71	0.92	1.03	1.13	0.96
Gain over r = 5 (dB)	0.23	0.12	0.10	0.13	0.21	0.18

Table 2. NRMSE values at several acceleration (undersampling) factors for the L+S method [30] and for the algorithm for (P2) with $r = 5$ (full rank), $r = 1$ (DINO-KAT MRI) and fixed D (DCT) cases. The best NRMSE values for each acceleration factor are marked in bold, and the improvements (gain) provided by DINO-KAT MRI are indicated in decibels (dB).

Table 2 lists the NRMSE values for conventional L+S [30] and the proposed DINO-KAT MRI ($r = 1$ and 64×5 (space-time) reshaped atoms) method at various undersampling factors. The NRMSEs achieved by the algorithm for (P2) with fixed D (DCT) and for the adaptive $r = 5$ (full rank) case are also shown. DINO-KAT MRI with rank-1 atoms provides the best reconstruction errors for each undersampling factor tested. In particular, it provides improvements up to 1.13 dB over the L+S method and up to 0.23 dB over the full rank $r = 5$ case. Figure 3 shows two representative frames of the DINO-KAT MRI reconstruction from 8x undersampled data. The reconstructed frames are visually very close to the reference frames.

4. CONCLUSIONS

This paper investigated a novel framework for structured dictionary learning with low rank (reshaped) atoms. We adopted an efficient algorithm for ℓ_0-based dictionary learning and presented theoretical and empirical convergence results for the method. Our experiments showed the promise of our schemes in inverse problem settings. In general, the effectiveness of the low rank atom model in applications would depend on the properties of the specific underlying data.

5. REFERENCES

[1] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[2] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE International Conference on Computer Vision*, Sept 2009, pp. 2272–2279.

[4] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1028–1041, 2011.

[5] 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.

[6] R. Gribonval and K. Schnass, "Dictionary identification–sparse matrix-factorization via l_1 -minimization," *IEEE Trans. Inform. Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.

[7] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2055–2065, 2013.

[8] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2010*, 2010, pp. 3501–3508.

[9] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

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

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

[13] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[14] L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 79–82, Jan 2013.

[15] A. Rakotomamonjy, "Direct optimization of the dictionary learning problem," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5495–5506, 2013.

[16] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," in *Proceedings of the 25th Annual Conference on Learning Theory*, 2012, pp. 37.1–37.18.

[17] S. Arora, R. Ge, and A. Moitra, "New algorithms for learning incoherent and overcomplete dictionaries," 2013, Preprint: http://arxiv.org/pdf/1308.6273v5.pdf.

[18] C. Bao, H. Ji, Y. Quan, and Z. Shen, "L0 norm based dictionary learning by proximal methods with global convergence," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3858–3865.

[19] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon, "Learning sparsely used overcomplete dictionaries," *Journal of Machine Learning Research*, vol. 35, pp. 1–15, 2014.

[20] S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Efficient sum of outer products dictionary learning (SOUP-DIL) and its application to inverse problems," 2016, Submitted.

[21] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Trans. on Signal Process.*, vol. 62, no. 4, pp. 883–891, 2014.

[22] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[23] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," http://www.cs.technion.ac.il/~ronrubin/Publications/KSVD-OMP-v2.pdf, 2008, Technion - Computer Science Department - Technical Report.

[24] R. T. Rockafellar and Roger J.-B. Wets, *Variational Analysis, 1st edn.*, Springer-Verlag, 1997.

[25] L. Howard, "Dartmouth college electron microscope facility - electron microscope images," http://remf.dartmouth.edu/miscellaneous_SEM_P1/, 2011, [Online; accessed Dec. 2015].

[26] M. Elad, "Michael Elad personal page," http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip, 2009, [Online; accessed Nov. 2015].

[27] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[28] S. Ravishankar and Y. Bresler, "Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2519–2557, 2015.

[29] H. S. Anderson, J. Ilic-Helms, B. Rohrer, J. Wheeler, and K. Larson, "Sparse imaging for fast electron microscopy," in *Proc. SPIE*, 2013, vol. 8657, pp. 86570C–86570C–12.

[30] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[31] R. Otazo, "L+S reconstruction matlab code," http://cai2r.net/resources/software/ls-reconstruction-matlab-code, 2014, [Online; accessed Mar. 2016].