DOUBLY SPARSE TRANSFORM LEARNING WITH CONVERGENCE GUARANTEES

Saiprasad Ravishankar and Yoram Bresler

Department of Electrical and Computer Engineering and the Coordinated Science Laboratory,
University of Illinois, Urbana-Champaign, IL 61801, USA

ABSTRACT

The sparsity of natural signals in transform domains such as the DCT has been heavily exploited in various applications. Recently, we introduced the idea of learning sparsifying transforms from data, and demonstrated the usefulness of learnt transforms in image representation, and denoising. However, the learning formulations therein were non-convex, and the algorithms lacked strong convergence properties. In this work, we propose a novel convex formulation for square sparsifying transform learning. We also enforce a doubly sparse structure on the transform, which makes its learning, storage, and implementation efficient. Our algorithm is guaranteed to converge to a global optimum, and moreover converges quickly. We also introduce a non-convex variant of the convex formulation, for which the algorithm is locally convergent. We show the superior promise of our learnt transforms as compared to analytical sparsifying transforms such as the DCT for image representation.

***Index Terms*—** Sparse representations, Convex learning

1. INTRODUCTION

Sparse representation of signals has become very popular in recent years. Various sparse models have been studied such as the *synthesis model* [1], *analysis model* [1, 2], and *transform model* [3].

In this work, we focus on the transform model which suggests that a signal $y \in \mathbb{R}^n$ is approximately sparsifiable using a transform $W \in \mathbb{R}^{m\times n}$, i.e., $Wy = x + \eta$, where x is sparse in some sense, and η is a small residual in the *transform domain*. Natural signals are known to be approximately sparse in analytical transform domains such as Wavelets [4]. The transform model is more general than the analysis model [3]. Moreover, it allows for much faster computations than the synthesis and analysis models. When a sparsifying transform W is known for a signal y, the process of obtaining a sparse code x of sparsity s is called transform sparse coding [3], and involves minimizing $\|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$. This is an easy problem whose solution is obtained exactly by zeroing out all but the s coefficients of largest magnitude in Wy. In contrast, sparse coding with either the synthesis [5, 6, 7], or analysis [8, 2, 9, 10] dictionaries involves solving an NP-hard problem [11, 12] approximately. Given W and sparse code x, one can recover a least squares estimate of y by minimizing $\|Wy - x\|_2^2$ over all $y \in \mathbb{R}^n$. The recovered signal is $W^\dagger x$, where $W^\dagger$ is the pseudo-inverse of W.

Adapting a dictionary or transform to data can be advantageous in various applications [13, 14]. While the idea of learning a synthesis [15, 16, 17] or analysis [18, 19, 8, 20] dictionary has received recent attention, these formulations are typically non-convex and NP-hard, and the approximate algorithms are still computationally expensive. In this paper, we focus instead on the learning of *square* sparsifying transforms $W \in \mathbb{R}^{n\times n}$. Given a matrix $Y \in \mathbb{R}^{n\times N}$ whose columns represent training signals, we recently proposed learning a square 'unstructured' transform W by minimizing the sparsification error $\|WY - X\|_F^2$, where X is the sparse code [3, 21]. We also introduced regularizers to enourage well-conditioning of W [3]. Unlike synthesis or analysis dictionary learning, the transform learning formulation [3] does not include a highly non-convex function involving the product of two unknown matrices. However, the formulation in [3] is still non convex, and the convergence of the algorithm therein is not proven. A chief advantage of transform learning is that it has a low computational cost, and is typically much faster than dictionary learning [3].

More recently, we explored the learning of doubly sparse transforms [22, 21] $W = B\Phi$, where $\Phi \in \mathbb{R}^{n\times n}$ is an analytical transform with an efficient implementation, and $B \in \mathbb{R}^{n\times n}$ is a sparse matrix. The structure $W = B\Phi$ is motivated by the fact that Φ matrices such as the DCT when applied to natural signals produce a result that is already approximately sparse. Thus, by further modifying the result using only a sparse B, one can produce a highly sparse result. Doubly sparse transforms can be learnt, stored, and implemented efficiently [21]. In fact, imposing the doubly sparse property leads to faster convergence of learning compared to the unstructured case [21]. We refer the reader to [21] for many other useful properties of doubly sparse learning. However, doubly sparse learning [21] similar to unstructured learning lacks strong convergence properties. Nevertheless, adaptive transforms (both doubly sparse and unstructured) have been shown to be useful in various applications [21, 23].

In this work, we propose a novel convex formulation for square doubly sparse transform learning. We also propose a non-convex formulation based on the convex one. Unlike prior work, we do provide strong convergence guarantees for our algorithms. We demonstrate the usefulness of our learning schemes for image representation.

2. TRANSFORM LEARNING

2.1. Problem Formulations

Given the training matrix $Y \in \mathbb{R}^{n\times N}$, we recently proposed to learn a square doubly sparse transform $W = B\Phi$ for the case of orthonormal Φ as follows [21].

$$\text{(P0)}\quad \min_{B,X} \|BZ - X\|_F^2 - \lambda \log |\det B| + \lambda \|B\|_F^2$$
$$s.t.\ \ \|B\|_0 \leq r,\ \|X_j\|_0 \leq s\ \ \forall\ j$$

Here, $Z = \Phi Y$, and the columns of $X \in \mathbb{R}^{n\times N}$ denote the sparse codes of the signals (columns) in Y. The subscript j indexes the j^{th} column, and the sparsity level allowed for each training signal is s. The term $\|BZ - X\|_F^2$ in (P0) is the sparsification error [3]. The $\log |\det B|$ penalty helps enforce full rank on the matrix B and eliminates degenerate solutions (e.g., with zero, or repeated rows). The $\|B\|_F^2$ penalty in (P0) helps remove a 'scale ambiguity' [3] in the solution, and together with the $-\log |\det B|$ penalty helps control the condition number of the learnt transform (cf. [3]).

This work was supported in part by the National Science Foundation (NSF) under grants CCF-1018660 and CCF-1320953.

Well-conditioned (but not necessarily unit-conditioned) transforms have been observed to perform well in applications [3, 21]. (Note that the condition number of B equals that of $B\Phi$ in the case of orthonormal Φ.) As $\lambda \to \infty$ in (P0), the condition number of the optimal/minimizing transform(s) tends to 1 [21]. The sparsity term $\|B\|_0$ is defined as $\sum_{i,j} 1_{\{B_{ij}\neq 0\}}$, with B_{ij} the entry of B from row i and column j, and $1_{\{B_{ij}\neq 0\}}$ the indicator function of $B_{ij} \neq 0$. The maximum allowed sparsity level for B is r.

Experimental results for (P0) [21] indicate that the learnt B for natural images has an interesting structure of a positive diagonal and an approximately skew-symmetric off-diagonal. This structure is observed with various analytical Φ such as the DCT, Hadamard, Karhunen-Loève Transform (KLT), etc. This observed structure is a motivation for our work here on convex learning.

We propose to model a sparse transform $B \in \mathbb{R}^{n\times n}$ as $I_n + A$, where I_n (or, simply I) is the $n \times n$ identity, and A is a skew-symmetric matrix satisfying $A^T = -A$ (skew-symmetry implies that A has a zero diagonal), with $(\cdot)^T$ denoting the matrix transpose operation. This B corresponds to a transform $W = B\Phi = \Phi + A\Phi$, which is the sum of the analytical Φ matrix and a deviation term $A\Phi$.

For the proposed B, we have $B^TB = (I-A)(I+A) = I - A^2$. If A is small (has small norm), then B is approximately orthonormal, since the second order deviation term A^2 above can be considered negligible. Thus, the condition number of B can be controlled simply by controlling the magnitude of A. Our convex formulation for doubly sparse transform learning is as follows.

$$\min_{A,X} \|(I+A)Z - X\|_F^2 + \frac{\eta}{4}\left\|A + A^T\right\|_F^2 + \mu\|A\|_1 + \xi\|X\|_1$$
$$s.t.\ \ A_{ii} = 0\ \forall\ i \qquad \text{(P1)}$$

Here, η, μ, and ξ are non-negative weights. The (trivial) condition $A_{ii} = 0\ \forall i$, although written as a constraint for simplicity, is in fact hard coded into the objective function, i.e., the optimization is only performed over the off-diagonal elements of A. Note that here $B = I_n + A$, where A has zeros on the diagonal, and is assumed to be approximately skew-symmetric. Approximate rather than exact skew-symmetry was observed in [21], and leads to slightly better performance in our experiments. Since A can be written as the sum of its orthogonal symmetric and skew-symmetric parts, i.e., $A = \frac{A+A^T}{2} + \frac{A-A^T}{2}$, the penalty $\frac{1}{4}\left\|A+A^T\right\|_F^2$ in (P1) helps ensure that the energy in the symmetric part is sufficiently small. The penalty $\|A\|_1 = \sum_{i,j}|A_{ij}|$ is to enforce sparsity of the off-diagonal of A. It also serves to keep A (magnitude) small, so that B is approximately orthonormal (i.e., is well-conditioned). Similarly, $\|X\|_1 = \sum_i \|X_i\|_1$ ensures sparsity of the columns of X. One could alternatively replace the penalty $\xi\|X\|_1$ with $\sum_i \xi_i \|X_i\|_1$, when appropriate weights ξ_i are known. The penalty $\|(I+A)Z - X\|_F^2$ in (P1) measures the sparsification error.

(P1) is a linear least squares problem in X and the off-diagonal of A, with additional ℓ_1 norm regularizers, and is therefore, convex. We believe this is the first convex sparsifying transform learning formulation. However, the quadratic part of the cost in (P1) is not strictly convex, since it has a linear variety of minimizers $(\hat{A}, \hat{X})$ satisfying $\hat{X} = (I+\hat{A})Z$, and $\hat{A} = -\hat{A}^T$. Thus, the ℓ_1 regularizers in (P1) can help ensure that the optimal minimizer(s) is sparse.

We also propose the following non-convex variant of Problem (P1), where the ℓ_1 penalty on X is replaced by an ℓ_0 constraint.

$$\text{(P2)}\ \ \min_{A,X} \|(I+A)Z - X\|_F^2 + \frac{\eta}{4}\left\|A + A^T\right\|_F^2 + \mu\|A\|_1$$
$$s.t.\ \ A_{ii} = 0\ \forall\ i,\ \ \|X_j\|_0 \leq s\ \ \forall\ j$$

Problem (P2) has fewer aspects of non-convexity than (P0), since it lacks the log-determinant penalty and the ℓ_0 constraint on B.

Both Problems (P1) and (P2) have an analytical solution for X for fixed A [3]. In the case of (P1), the solution is given as $X = S_{\xi/2}(Z + AZ)$, where $S_{\xi/2}(\cdot)$ is the soft-thresholding operator. For a matrix C, $S_{\xi/2}(C) = \text{sign}(C) \odot (|C| - \xi/2)_+$, where "$\odot$" represents element-wise multiplication between matrices, $\text{sign}(\cdot)$ provides the signs of the elements of a matrix, and $(\cdot)_+$ zeros out all but the non-negative elements of a matrix. In the case of (P2), the solution for X with fixed A is obtained by zeroing out all but the s coefficients of largest magnitude in each column of $Z + AZ$ (when this solution is unique, it is equivalent to hard-thresholding $Z + AZ$ with a possibly different threshold for each column). The solutions for X with a fixed A in (P1) and (P2) are thus non-identical (except in extreme cases when they are both either 0 (for sufficiently large ξ and $s = 0$), or $Z + AZ$ (when $\xi = 0, s = n$)), since soft-thresholding always causes shrinkage of large coefficients, while hard-thresholding does not have such an effect. However, despite their non-equivalence, both (P1) and (P2) perform well and quite similarly in practice.

2.2. Algorithms and Properties

DOSLIST Algorithm. Our algorithm for Problem (P1) is a scale-invariant version of standard FISTA [24] that uses multiple Lipschitz constants. We define $f(A, X) = \|(I+A)Z - X\|_F^2 + \frac{\eta}{4}\left\|A + A^T\right\|_F^2$, where A is assumed to be zero on the main diagonal. Then, for any set of matrices A, A', X, X' (of appropriate sizes, and with A, A' having zero diagonals), the function f satisfies the following inequality for appropriate constants L_A and L_X.

$$f(A', X') \leq \left\langle \nabla_A f(A, X), A' - A \right\rangle + \frac{L_A}{2}\left\|A' - A\right\|_F^2 \qquad (1)$$
$$+ f(A, X) + \left\langle \nabla_X f(A, X), X' - X \right\rangle + \frac{L_X}{2}\left\|X' - X\right\|_F^2$$

Here, $\nabla_X f$ and $\nabla_A f$, respectively, denote the gradients (arranged in matrix form) of f with respect to X and to the off-diagonal elements of A, with the diagonal of $\nabla_A f$ fixed to zero. Specifically, $\nabla_A f(A, X) = G \odot (2(I+A)ZZ^T - 2XZ^T + \eta A + \eta A^T)$ with G a matrix of all ones and a zero main diagonal, and $\nabla_X f(A, X) = 2(X - Z - AZ)$. For matrices Q and R, the inner product $\langle \cdot, \cdot \rangle$ in (1) is defined by $\langle Q, R \rangle = \text{trace}(R^T Q)$. Equation (1) is obviously satisfied when $L_A = L_X = L$, where L is a 'global' Lipschitz constant [24] of ∇f. However, L_A and L_X need not coincide in general (e.g., (1) may hold with $L_X \ll L_A = L$).

Our algorithm for solving (P1) called DOSLIST is presented in Fig. 1. It is similar to FISTA, but uses the block constants L_A and L_X. While we use constant stepsizes here, one can also obtain a version of DOSLIST with backtracking (similar to FISTA [24]).

DOSLAM Algorithm. For the non-convex Problem (P2), we propose an alternating algorithm similar to the one previously proposed for Problem (P0) [21]. We call our proposed algorithm DOubly Sparse Learning by Alternating Minimization (DOSLAM). In one step of the algorithm called the *Sparse coding step*, we solve (P2) with fixed A.

$$\min_{X} \|(I+A)Z - X\|_F^2\ \ s.t.\ \ \|X_j\|_0 \leq s\ \ \forall\ j \qquad (2)$$

The solution X is obtained exactly as $X = H_s(Z + AZ)$, where the operator $H_s(\cdot)$ zeros out all but the s coefficients of largest magnitude in each column of a given matrix. In the other step of DOSLAM called the *Transform update step*, we solve (P2) with fixed X.

$$\min_{A; A_{ii}=0\,\forall i} \|(I+A)Z - X\|_F^2 + \frac{\eta}{4}\left\|A + A^T\right\|_F^2 + \mu\|A\|_1 \quad (3)$$

```
Input : $Z = \Phi Y$ - Data, $L_A$, $L_X$ - Constants satisfying equation (1), $J$ - number of iterations.
Initialization : $Q^1 = A^0, R^1 = S_{\xi/2}(Z+A^0 Z) = X^0, t_1 = 1.$
For k = 1:J Repeat
    $A^k = S_{\mu/L_A}\left(Q^k - \frac{1}{L_A}\nabla_A f(Q^k, R^k)\right)$
    $X^k = S_{\xi/L_X}\left(R^k - \frac{1}{L_X}\nabla_X f(Q^k, R^k)\right)$
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $Q^{k+1} = A^k + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(A^k - A^{k-1}\right)$
    $R^{k+1} = X^k + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(X^k - X^{k-1}\right)$
End
```

Fig. 1. DOubly Sparse Learning by Iterative Soft Thresholding (DOSLIST) Algorithm for Problem (P1). The initial A^0 has a zero diagonal, and the update of A^k above keeps the diagonal as 0.

Here, as before, the condition $A_{ii} = 0 \; \forall \, i$, is directly incorporated into the objective, making the above problem unconstrained. The objective function here is convex, and is similar to the objective of (P1). Thus, it can be minimized using iterative algorithms like ISTA [25, 24], or MFISTA [26] (which is the monotone version of FISTA). Both ISTA and MFISTA work well for this step. However, we employ ISTA here to obtain strong convergence results for DOSLAM. Note that ISTA is guaranteed to converge to a global minimizer of the objective in (3) at $O(1/k)$ rate [24]. ISTA requires a Lipschitz constant $\hat{L}$ as input. Since X is fixed in (3), we are only interested in the Lipschitz constant of $\nabla_A f(A, X)$ (fixed X).

Convergence of DOSLIST. We denote the full objective function of (P1) as $F(A, X)$, where A has a zero diagonal. The following theorem provides the $O(1/k^2)$ convergence rate of DOSLIST.

Theorem 1. *Let $\{A^k\}$, $\{X^k\}$, $\{Q^k\}$, $\{R^k\}$ denote the iterate sequences generated by the DOSLIST algorithm for data Z. Further, let (A^*, X^*) denote any minimizer of Problem (P1). Then, for any $k \geq 1$, we have*

$$F(A^k, X^k) - F(A^*, X^*) \leq \frac{2C}{(k+1)^2} \tag{4}$$

where the constant $C = L_A \left\|A^0 - A^*\right\|_F^2 + L_X \left\|X^0 - X^*\right\|_F^2$.

The proof of the above theorem follows that of FISTA [24] closely, except that expressions in [24] involving a single Lipschitz constant are broken into their block-Lipschitz components. The full details of the proof will be presented elsewhere [27].

We now demonstrate the interesting scale-invariance behavior of the DOSLIST algorithm. We first show how Problem (P1), and the constants L_A and L_X in (1) change when the data Z is scaled. For our analysis, we introduce Z into our previous notation and write $f(A, X, Z) = \|(I + A)Z - X\|_F^2 + \frac{\eta}{4}\left\|A + A^T\right\|_F^2$. Similarly, the overall objective is $F(A, X, Z)$.

Let us call the objective of (P1) with data Z and weights η, μ, ξ, i.e., $F(A, X, Z)$, the 'un-scaled' objective. When Z is scaled by a non-zero scalar $\alpha \in \mathbb{R}$ in (P1), we also need to scale the weights η and μ by α^2, and ξ by $|\alpha|$. Then, by making the substitution $X = \alpha X'$, the new objective $\widetilde{F}$ with αZ and scaled weights satisfies $\widetilde{F}(A, X, \alpha Z) = \alpha^2 F(A, X', Z)$. Therefore, both F and $\widetilde{F}$ have the same set of minimizers with respect to A. Moreover, the minimizers with respect to X for $\widetilde{F}$ are α times the corresponding minimizers for F (this makes perfect sense since the data Z was trivially scaled).

The following lemma (which we provide without proof) shows the behavior of the constants L_A and L_X with scaling.

Lemma 1. *Let L_A and L_X be constants satisfying* (1) *for the function $f(A, X, Z)$ with data Z and weight η. For any non-zero scalar $\alpha \in \mathbb{R}$, if we replace Z with αZ, and η with $\alpha^2 \eta$, then the modified function $\widetilde{f}$ satisfies* (1) *with modified constants*

$$\tilde{L}_A = \alpha^2 L_A, \; \tilde{L}_X = L_X. \tag{5}$$

We can also write the constants as $L_A = C_1 \sigma_1^2$, $L_X = C_2$, where σ_1 is the largest singular value of Z, and C_1, C_2 are the constants satisfying (1) when Z is scaled to have unit spectral norm. This form of L_A and L_X models their behavior with respect to scaling of Z, as dictated by Lemma 1.

We now use Lemma 1 to show that the DOSLIST algorithm is scale-invariant. Let $\{A^k\}$ and $\{X^k\}$ be the DOSLIST iterate sequences obtained with the (un-scaled) input Z, weights η, μ, ξ, and constants L_A, L_X satisfying (1). Futher, let $\{A_1^k\}$ and $\{X_1^k\}$ denote the modified sequences generated by the DOSLIST algorithm with input αZ, and appropriately scaled weights in (P1), and appropriately scaled constants. Then, by looking at the effect of the scaling on each step in Figure 1, it is easy to observe that $A_1^k = A^k$ and $X_1^k = \alpha X^k$. Thus, the DOSLIST algorithm always generates the same sequence $\{A^k\}$ irrespective of the scaling on Z. It can also be shown that scaling Z by α in (P1) (along with scaling of the weights) simply causes equation (4) of Theorem 1 to scale by α^2 throughout. If we divide (4) by the non-negative $F(A^*, X^*)$ (assuming it is non-zero), then the resulting equation is scale-invariant.

We now show that the standard FISTA [24] is scale-dependant. Standard FISTA is equivalent to using a single $L = \max(L_A, L_X)$ (this is the smallest L for which equation (1) becomes equal to the corresponding condition for f in FISTA [24]) in DOSLIST. For example, when $\sigma_1(Z)$ (the scaling) is sufficiently large, then $L = \max(C_1 \sigma_1^2, C_2) = C_1 \sigma_1^2$, and it is easy to see that the steps of standard FISTA (Fig. 1 with $L_X = L_A = L$) are not scale-invariant in this setting (i.e., if a particular large choice of σ_1 gets scaled (e.g., by 2), it results in a totally new/unrelated iterate sequence). Moreover, it can be shown for this case that the bound in Theorem 1 (with single L) is also not homogeneous to scaling, and the constant C scales badly as $O(\sigma_1^4)$. In practice, we observed that standard FISTA (with either constant step size or backtracking) has a poor (slow) convergence behavior for (P1), unless the scaling of Z is manually tuned for better convergence. Thus, the DOSLIST algorithm with the scale-invariant behavior eliminates a need for scale tuning.

Convergence of DOSLAM. Problem (P2) has the constraint $\|X_j\|_0 \leq s \, \forall j$, which can instead (equivalently) be added as a penalty in the objective by using a barrier function $v(X)$ (taking the value $+\infty$ when the constraint is violated, and zero otherwise). In this form, Problem (P2) is unconstrained (note no optimization over diagonal of A), and we denote its objective as $g(A, X)$. For a vector u, let $\beta_j(u)$ denote the magnitude of the j^{th} largest element (magnitude-wise) of u. We then have the following Theorem on the convergence of our algorithm for (P2).

Theorem 2. *Let $\{A^k, X^k\}$ denote the iterate sequence generated by the DOSLAM algorithm for (P2) with data Z and initialization (A^0, X^0). Then, the objective sequence $\{g(A^k, X^k)\}$ is monotone decreasing, and converges to a finite value, say g^*. Moreover, the iterate sequence is bounded, and every accumulation point (A, X) of the iterate sequence is a fixed point of the algorithm satisfying the following local optimality condition.*

$$g(A + dA, X + dX) \geq g(A, X) = g^* \tag{6}$$

The condition holds for all $dA \in \mathbb{R}^{n \times n}$ with a zero diagonal, and all $dX \in \mathbb{R}^{n \times N}$ in the half-space $\langle (I + A)Z - X, dX \rangle \leq 0$. Furthermore, the condition also holds for dX in the local region defined by

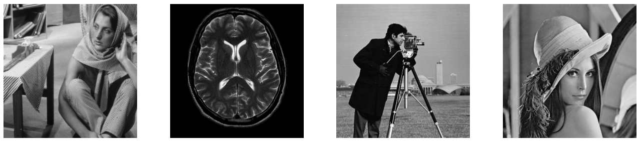

Fig. 2. The test images - Barbara, brain [14], Cameraman, and Lena, which we label with numbers 1 through 4, respectively.

$\|dX\|_\infty \triangleq \max_{i,j} |dX_{ij}| < \min_j \{\beta_s(U_j) : \|U_j\|_0 > s\}$, *where* $U = (I + A)Z$. *If* $\|U_j\|_0 \leq s \,\forall\, j$, *then* dX *can be arbitrary.*

Theorem 2 indicates local convergence of our alternating algorithm. Every accumulation point (A, X) of the DOSLAM iterate sequence is a local optimum by (6), and satisfies $g(A, X) = g^*$. Thus, all the accumulation points are equivalent (in terms of their cost), or equally good local minima. We can therefore say that the objective converges to a local minimum for DOSLAM. Equation (6) holds not only for local (small) perturbations in X, but also for arbitrarily large perturbations of X in a half space. Furthermore, (6) holds for the algorithm irrespective of initialization. However, the local minimum g^* to which the cost converges may possibly depend on initialization. The condition (6) also holds irrespective of the number of iterations of ISTA in the transform update step. All these properties and the large set of permissible perturbations (dA, dX) in Theorem 2 indicate a strong ('almost global') convergence for our DOSLAM algorithm. For reasons of space, the proof of Theorem 2 is presented elsewhere [27].

The proposed algorithms for both (P1) and (P2) have a low computational cost, which scales in order as $O(n^2N)$ per-iteration. The algorithms involve operations with sparse matrices, which can be implemented very efficiently (similar to [21]).

3. NUMERICAL EXPERIMENTS

We study the usefulness of the proposed transform learning schemes for representing the four 512 × 512 labeled images in Fig. 2. We learn $W = B\Phi$ from the 12 × 12 (zero mean) non-overlapping patches of the images, with Φ being the patch-based 2D DCT [3]. We compare the transforms learnt via (P1) and (P2), with those learnt via (P0), and the fixed patch-based 2D DCT itself. The parameters for (P1) and (P2) are $n = 144$, $\eta = (3.26 \times 10^{-3})\sigma_1^2(Z)$, $\mu = (2.18 \times 10^{-6})\sigma_1^2$, $s = 24$, $A^0 = 0$. We use a different $\xi_i = 0.44 \times \beta_{s+1}(Z_i)$ (scales linearly with $Z = \Phi Y$ assuming $\beta_{s+1}(Z_i) \neq 0$ $\forall\, i$) for each patch (column of X) in (P1). For DOSLIST, $L_A = 2.56\sigma_1^2$ and $L_X = 3.7$, which we found empirically. For DOSLAM, we use $\hat{L} = 1.42\sigma_1^2$ (in ISTA), and run 100 iterations of ISTA in the transform update step. For (P0), $\lambda = 8.7 \times 10^{-3}\sigma_1^2$, $r = 0.25 \times n^2$, with n, s the same as before. We stop the iterations of the algorithms when the relative iterate change [21] falls below 0.01%. We also set a maximum iteration count of 300.

Fig. 3 shows the evolution of the objective over iterations for our algorithms for (P1) and (P2), for the cameraman image. The objectives converge quickly for our algorithms. The magnitudes of the A matrices learnt via (P1) and (P2) are also shown in Fig. 3. They appear sparse, and are similar.

For (P1) and (P2), we generate exactly sparse transforms at a sparsity level of $0.25 \times n^2$ by thresholding the learnt $B = I_n + A$. Note that the transforms learnt via (P1) and (P2) are already exactly sparse. However, they typically also contain many elements close to zero, which can be thresholded, without affecting the transform quality, but improving its sparsity.

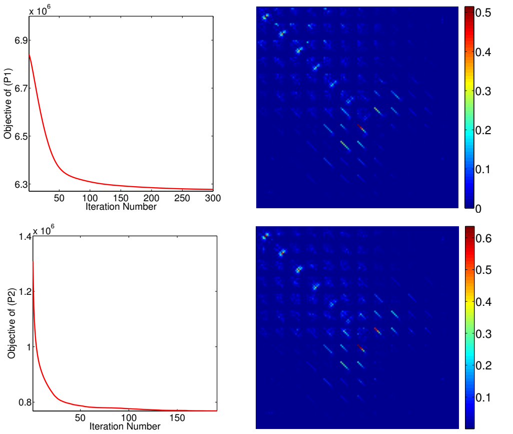

Fig. 3. The evolution of the objectives of (P1) (top left) and (P2) (bottom left) for Cameraman, and the magnitude of the corresponding learnt A for (P1) (top right) and (P2) (bottom right).

Image	NSE				rPSNR			
	P1	P2	P0	DCT	P1	P2	P0	DCT
1	3.6	3.0	2.8	4.4	34.99	35.32	35.59	33.76
2	0.5	0.4	0.4	0.6	45.32	45.72	45.77	44.12
3	0.6	0.5	0.4	1.1	43.34	44.19	44.39	40.27
4	3.0	2.6	2.4	3.1	37.74	37.97	38.13	37.13

Table 1. NSE (in percentage), and rPSNR (dB) for various images obtained using our algorithms for (P1) and (P2), along with the corresponding values for (P0) [21], and the fixed DCT.

We measure the quality of the transforms $W = B\Phi$ using the normalized sparsification error (NSE), and recovery peak signal to noise ratio (rPSNR) metrics. The NSE metric [3] is defined as $\|WY - X\|_F^2 / \|WY\|_F^2$, where $X = H_s(WY)$, and it measures the fraction of energy lost in sparse fitting in the transform domain. The rPSNR metric is defined as $255\sqrt{P} / \left\|Y - W^{-1}X'\right\|_F$ in dB, where P is the number of image pixels. It measures the error in recovering the patches Y (or, the image for non-overlapping patches) as $W^{-1}X'$ from the sparse codes X'. While we can use X' obtained as just $H_s(WY)$ [3], we found that the rPSNR improves by setting only the support of X' to the support of $H_s(WY)$, and then performing a simple least squares update of the nonzero elements of X' to minimize $\left\|Y - W^{-1}X'\right\|_F$. rPSNR is a simple surrogate for the compression performance of transforms.

Table 1 provides the NSE and rPSNR values for the transforms learnt via various algorithms, along with the corresponding values for the 2D DCT Φ. The learnt transforms using the proposed algorithms (which are all well-conditioned) for (P1) and (P2) are seen to provide much better recovery and sparsification compared to the analytical DCT. Moreover, they tend to perform comparably to the transforms learnt via (P0). The performance for (P2) is slightly better than for the fully convex (P1), due to the non-equivalence of the problem formulations, discussed in Section 2.1.

4. CONCLUSIONS

In this paper, we presented the first convex sparsifying transform learning formulation, and an algorithm guaranteeing $O(1/k^2)$ convergence to a global optimum. We also presented a non convex formulation for which our algorithm has local convergence. Our learnt transforms provide much better sparsification errors and recovery PSNRs than analytical transforms. They also perform comparably to those learnt using previous non convex (non-guaranteed) schemes in our experiments. The usefulness of the proposed schemes in denoising [21] and other applications [23] merits further study.

5. REFERENCES

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

[2] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, “Cosparse analysis modeling - uniqueness and algorithms,” in *ICASSP*, 2011, pp. 5804–5807.

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

[4] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[5] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition,” in *Asilomar Conf. on Signals, Systems and Comput.*, 1993, pp. 40–44 vol.1.

[6] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Trans. Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[8] R. Rubinstein, T. Faktor, and M. Elad, “K-SVD dictionary-learning for the analysis sparse model,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5405–5408.

[9] A. Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vis.*, vol. 20, no. 1-2, pp. 89–97, 2004.

[10] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, “Noise aware analysis operator learning for approximately cosparse signals,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5409–5412.

[11] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[12] G. Davis, S. Mallat, and M. Avellaneda, “Adaptive greedy approximations,” *Journal of Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[13] 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.

[14] 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.

[15] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[16] K. Engan, S. O. Aase, and J. Hakon Husoy, “Method of optimal directions for frame design,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, pp. 2443–2446.

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

[18] G. Peyré and J. Fadili, “Learning analysis sparsity priors,” in *Proc. Int. Conf. Sampling Theory Appl. (SampTA)*, Singapore, May 2-6, 2011.

[19] M. Yaghoobi, S. Nam, R. Gribonval, and M. Davies, “Analysis operator learning for overcomplete cosparse representations,” in *European Signal Processing Conference (EUSIPCO)*, 2011, pp. 1470–1474.

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

[21] S. Ravishankar and Y. Bresler, “Learning doubly sparse transforms for images,” *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4598–4612, 2013.

[22] S. Ravishankar and Y. Bresler, “Learning doubly sparse transforms for image representation,” in *IEEE Int. Conf. Image Process.*, 2012, pp. 685–688.

[23] S. Ravishankar and Y. Bresler, “Sparsifying transform learning for compressed sensing MRI,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2013, pp. 17–20.

[24] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[25] A. Chambolle, R. A. De Vore, Nam-Yong Lee, and B. J. Lucier, “Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 319–335, 1998.

[26] A. Beck and M. Teboulle, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, 2009.

[27] S. Ravishankar and Y. Bresler, “Fast doubly sparse transform learning with convergence guarantees,” 2014, to be submitted.