Published in final edited form as:
IEEE Trans Comput Imaging. 2017 December ; 3(4): 694–709. doi:10.1109/TCI.2017.2697206.

Efficient Sum of Outer Products Dictionary Learning (SOUP-DIL) and Its Application to Inverse Problems

Saiprasad Ravishankar [Member, IEEE], **Raj Rao Nadakuditi [Member, IEEE]**, and **Jeffrey A. Fessler [Fellow, IEEE]**
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109 USA

Abstract

The sparsity of signals in a transform domain or dictionary has been exploited in applications such as compression, denoising and inverse problems. More recently, data-driven adaptation of synthesis dictionaries has shown promise compared to analytical dictionary models. However, dictionary learning problems are typically non-convex and NP-hard, and the usual alternating minimization approaches for these problems are often computationally expensive, with the computations dominated by the NP-hard synthesis sparse coding step. This paper exploits the ideas that drive algorithms such as K-SVD, and investigates in detail efficient methods for aggregate sparsity penalized dictionary learning by first approximating the data with a sum of sparse rank-one matrices (outer products) and then using a block coordinate descent approach to estimate the unknowns. The resulting block coordinate descent algorithms involve efficient closed-form solutions. Furthermore, we consider the problem of dictionary-blind image reconstruction, and propose novel and efficient algorithms for adaptive image reconstruction using block coordinate descent and sum of outer products methodologies. We provide a convergence study of the algorithms for dictionary learning and dictionary-blind image reconstruction. Our numerical experiments show the promising performance and speedups provided by the proposed methods over previous schemes in sparse data representation and compressed sensing-based image reconstruction.

Index Terms

Sparsity; Dictionary learning; Inverse problems; Compressed sensing; Fast algorithms; Convergence analysis

I. Introduction

The sparsity of natural signals and images in a transform domain or dictionary has been exploited in applications such as compression, denoising, and inverse problems. Well-known models for sparsity include the synthesis, analysis [1], [2], and transform [3], [4] (or

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes proofs and additional experimental results. Contact fessler@umich.edu for further questions about this work.

generalized analysis) models. Alternative signal models include the balanced sparse model for tight frames [5], where the signal is sparse in a synthesis dictionary and also approximately sparse in the corresponding transform (transpose of the dictionary) domain, with a common sparse representation in both domains. These various models have been exploited in inverse problem settings such as in compressed sensing-based magnetic resonance imaging [5]–[7]. More recently, the data-driven adaptation of sparse signal models has benefited many applications [4], [8]–[18] compared to fixed or analytical models. This paper focuses on data-driven adaptation of the synthesis model and investigates highly efficient methods with convergence analysis and applications, particularly inverse problems. In the following, we first briefly review the topic of synthesis dictionary learning before summarizing the contributions of this work.

A. Dictionary Learning

The well-known synthesis model approximates a signal $\mathbf{y} \in \mathbb{C}^n$ by a linear combination of a small subset of atoms or columns of a dictionary $\mathbf{D} \in \mathbb{C}^{n \times J}$, i.e., $\mathbf{y} \approx \mathbf{D}\mathbf{x}$ with $\mathbf{x} \in \mathbb{C}^J$ sparse, i.e., $\| \mathbf{x} \|_0 \ll n$. Here, the ℓ_0 "norm" counts the number of non-zero entries in a vector, and we assume $\| \mathbf{x} \|_0$ is much lower than the signal dimension n. Since different signals may be approximated using different subsets of columns in the dictionary **D**, the synthesis model is also known as a union of subspaces model [19], [20]. When $n = J$ and **D** is full rank, it is a basis. Else when $J > n$, **D** is called an overcomplete dictionary. Because of their richness, overcomplete dictionaries can provide highly sparse (i.e., with few non-zeros) representations of data and are popular.

For a given signal **y** and dictionary **D**, finding a sparse coefficient vector **x** involves solving the well-known synthesis *sparse coding* problem. Often this problem is to minimize $\|\mathbf{y} - \mathbf{D}x\|_2^2$ subject to $\| \mathbf{x} \|_0 \leq s$, where s is a set sparsity level. The synthesis sparse coding problem is NP-hard (Non-deterministic Polynomial-time hard) in general [21]. Numerous algorithms [22]–[27] including greedy and relaxation algorithms have been proposed for such problems. While some of these algorithms are guaranteed to provide the correct solution under certain conditions, these conditions are often restrictive and violated in applications. Moreover, these algorithms typically tend to be computationally expensive for large-scale problems.

More recently, data-driven adaptation of synthesis dictionaries, called dictionary learning, has been investigated [12], [28]–[31]. Dictionary learning provides promising results in several applications, including in inverse problems [8], [9], [13], [32]. Given a collection of signals $\{\mathbf{y}_i\}_{i=1}^N$ (e.g., patches extracted from some images) that are represented as columns of the matrix $\mathbf{Y} \in \mathbb{C}^{n \times N}$, the dictionary learning problem is often formulated as follows [30]:

$$\min_{\mathbf{D},\mathbf{X}} \|\boldsymbol{Y} - \mathbf{D}\mathbf{X}\|_F^2 \text{ s.t. } \|x_i\|_0 \leq s \; \forall i, \; \|\mathbf{d}_j\|_2 = 1 \forall j. \qquad \text{(P0)}$$

Here, $\mathbf{d}_j$ and $\mathbf{x}_i$ denote the columns of the dictionary $\mathbf{D} \in \mathbb{C}^{n \times J}$ and sparse code matrix $\mathbf{X} \in \mathbb{C}^{J \times N}$, respectively, and s denotes the maximum sparsity level (number of non-zeros in representations $\mathbf{x}_i$) allowed for each signal. Constraining the columns of the dictionary to have unit norm eliminates the scaling ambiguity [33]. Variants of Problem (P0) include replacing the ℓ_0 "norm" for sparsity with an ℓ_1 norm or an alternative sparsity criterion, or enforcing additional properties (e.g., incoherence [11], [34]) for the dictionary **D**, or solving an online version (where the dictionary is updated sequentially as new signals arrive) of the problem [12].

Algorithms for Problem (P0) or its variants [12], [29]–[31], [35]–[41] typically alternate in some form between a *sparse coding step* (updating **X**), and a *dictionary update step* (updating **D**). Some of these algorithms (e.g., [30], [38], [40]) also partially update **X** in the dictionary update step. A few recent methods update **D** and **X** jointly in an iterative fashion [42], [43]. The K-SVD method [30] has been particularly popular [8], [9], [13]. Problem (P0) is highly non-convex and NP-hard, and most dictionary learning approaches lack proven convergence guarantees. Moreover, existing algorithms for (P0) tend to be computationally expensive (particularly alternating-type algorithms), with the computations usually dominated by the sparse coding step.

Some recent works [41], [44]–[48] have studied the convergence of (specific) dictionary learning algorithms. However, these dictionary learning methods have not been demonstrated to be useful in applications such as inverse problems. Bao et al. [41] find that their proximal scheme denoises less effectively than K-SVD [8]. Many prior works use restrictive assumptions (e.g., noiseless data, etc.) for their convergence results.

Dictionary learning has been demonstrated to be useful in inverse problems such as in tomography [49] and magnetic resonance imaging (MRI) [13], [50]. The goal in inverse problems is to estimate an unknown signal or image from its (typically corrupted) measurements. We consider the following general regularized linear inverse problem:

$$\min_{\mathbf{y}\in\mathbb{C}^p} \|\mathbf{A}\mathbf{y}-\mathbf{z}\|_2^2+\zeta(\mathbf{y}) \qquad (1)$$

where $\mathbf{y} \in \mathbb{C}^p$ is a vectorized version of a signal or image (or volume) to be reconstructed, $\mathbf{z} \in \mathbb{C}^m$ denotes the observed measurements, and $\mathbf{A} \in \mathbb{C}^{m\times p}$ is the associated measurement matrix for the application. For example, in the classical denoising application (assuming i.i.d. gaussian noise), the operator **A** is the identity matrix, whereas in inpainting (i.e., missing pixels case), **A** is a diagonal matrix of zeros and ones. In medical imaging applications such as computed tomography or magnetic resonance imaging, the system operator takes on other forms such as a Radon transform, or a Fourier encoding, respectively. A regularizer $\zeta(\mathbf{y})$ is used in (1) to capture assumed properties of the underlying image **y** and to help compensate for noisy or incomplete data **z**. For example, $\zeta(\mathbf{y})$ could encourage the sparsity of **y** in some fixed or known sparsifying transform or dictionary, or alternatively, it could be an adaptive dictionary-type regularizer such as one based on (P0) [13]. The latter case corresponds to dictionary-blind image reconstruction, where the dictionary for the underlying image patches is unknown a priori. The goal is then to reconstruct both the image **y** as well as the dictionary **D** (for image patches) from the observed measurements **z**. Such an approach allows the dictionary to adapt to the underlying image [13].

B. Contributions

This work focuses on dictionary learning using a general overall sparsity penalty instead of column-wise constraints like in (P0). We focus on ℓ_0 "norm" penalized dictionary learning, but also consider alternatives. Similar to recent works [30], [51], we approximate the data (**Y**) by a sum of sparse rank-one matrices or outer products. The constraints and penalties in the learning problems are separable in terms of the dictionary columns and their corresponding coefficients, which enables efficient optimization. In particular, we use simple and exact block coordinate descent approaches to estimate the factors of the various rank-one matrices in the dictionary learning problem. Importantly, we consider the application of such sparsity penalized dictionary learning in inverse problem settings, and investigate the problem of overall sparsity penalized dictionary-blind image reconstruction. We propose novel methods for image reconstruction that exploit the proposed efficient dictionary learning methods. We provide a novel convergence analysis of the algorithms for overcomplete dictionary learning and dictionary-blind image reconstruction for both ℓ_0 and ℓ_1 norm-based settings. Our experiments illustrate the empirical convergence behavior of our methods, and demonstrate their promising performance and speed-ups over some recent related schemes in sparse data representation and compressed sensing-based [52], [53] image reconstruction. These experimental results illustrate the benefits of aggregate sparsity penalized dictionary learning, and the proposed ℓ_0 "norm"-based methods.

C. Relation to Recent Works

The sum of outer products approximation to data has been exploited in recent works [30], [51] for developing dictionary learning algorithms. Sadeghi et al. [51] considered a variation of the Approximate K-SVD algorithm [54] by including an ℓ_1 penalty for coefficients in the dictionary update step of Approximate K-SVD. However, a formal and rigorous description of the formulations and various methods for overall sparsity penalized dictionary learning, and their extensions, was not developed in that work. In this work, we investigate in detail Sum of OUter Products (SOUP) based learning methodologies in a variety of novel problem settings. We focus mainly on ℓ_0 "norm" penalized dictionary learning. While Bao et al. [41], [55] proposed proximal alternating schemes for ℓ_0 dictionary learning, we show superior performance (both in terms of data representation quality and runtime) with the proposed simpler direct block coordinate descent methods for sparse data representation. Importantly, we investigate the novel extensions of SOUP learning methodologies to inverse problem settings. We provide a detailed convergence analysis and empirical convergence studies for the various efficient algorithms for both dictionary learning and dictionary-blind image reconstruction. Our methods work better than classical overcomplete dictionary learning-based schemes (using K-SVD) in applications such as sparse data representation and magnetic resonance image reconstruction from undersampled data. We also show some benefits of the proposed ℓ_0 "norm"-based adaptive methods over corresponding ℓ_1 methods in applications.

D. Organization

The rest of this paper is organized as follows. Section II discusses the formulation for ℓ_0 "norm"-based dictionary learning, along with potential alternatives. Section III presents the dictionary learning algorithms and their computational properties. Section IV discusses the formulations for dictionary-blind image reconstruction, along with the corresponding algorithms. Section V presents a convergence analysis for various algorithms. Section VI illustrates the empirical convergence behavior of various methods and demonstrates their usefulness for sparse data representation and inverse problems (compressed sensing). Section VII concludes with proposals for future work.

II. Dictionary Learning Problem Formulations

This section and the next focus on the "classical" problem of dictionary learning for sparse signal representation. Section IV generalizes these methods to inverse problems.

A. ℓ_0 Penalized Formulation

Following [41], we consider a sparsity penalized variant of Problem (P0). Specifically, replacing the sparsity constraints in (P0) with an ℓ_0 penalty $\sum_{i=1}^{N} \|\mathbf{x}_i\|_0$ and introducing a variable $\mathbf{C} = \mathbf{X}^H \in \mathbb{C}^{N \times J}$, where $(\cdot)^H$ denotes matrix Hermitian (conjugate transpose), leads to the following formulation:

$$\min_{\mathbf{D},\mathbf{C}} \|\boldsymbol{Y} - \mathbf{D}\mathbf{C}^H\|_F^2 + \lambda^2 \|\mathbf{C}\|_0 \text{ s.t. } \|\mathbf{d}_j\|_2 = 1 \forall j. \qquad (2)$$

where $\|\mathbf{C}\|_0$ counts the number of non-zeros in matrix **C**, and λ^2 with $\lambda > 0$, is a weight to control the overall sparsity.

Next, following previous work like [30], [51], we express the matrix $\mathbf{DC}^H$ in (2) as a sum of (sparse) rank-one matrices or outer products $\sum_{j=1}^{J} \mathbf{d}_j \mathbf{c}_j^H$, where $\mathbf{c}_j$ is the jth column of **C**. This SOUP representation of the data **Y** is natural because it separates out the contributions of the various atoms in representing the data. For example, atoms of a dictionary whose contributions to the data (**Y**) representation error or modeling error are small could be dropped. With this model, (2) becomes (P1) as follows, where $\|\mathbf{C}\|_0 = \sum_{j=1}^{J} \|\mathbf{c}_j\|_0$:

$$\min_{\{\mathbf{d}_j,\mathbf{c}_j\}} \left\| \boldsymbol{Y} - \sum_{j=1}^{J} \mathbf{d}_j \mathbf{c}_j^H \right\|_F^2 + \lambda^2 \sum_{j=1}^{J} \|\mathbf{c}_j\|_0$$
$$\text{s.t. } \|\mathbf{d}_j\|_2 = 1,\ \|\mathbf{c}_j\|_\infty \le L \forall j. \qquad \text{(P1)}$$

As in Problem (P0), the matrix $\mathbf{d}_j \mathbf{c}_j^H$ in (P1) is invariant to joint scaling of $\mathbf{d}_j$ and $\mathbf{c}_j$ as $\alpha \mathbf{d}_j$ and $(1/\alpha)\mathbf{c}_j$, for $\alpha \neq 0$. The constraint $\|\mathbf{d}_j\|_2 = 1$ helps in removing this scaling ambiguity. We also enforce the constraint $\|\mathbf{c}_j\|_\infty \le L$, with $L > 0$, in (P1) [41] (e.g., $L = \|\mathbf{Y}\|_F$).

This is because the objective in (P1) is non-coercive. In particular, consider a dictionary **D** that has a column $\mathbf{d}_j$ that repeats. Then, in this case, the SOUP approximation for **Y** in (P1) could have both the terms $\mathbf{d}_j\mathbf{c}_j^H$ and $-\mathbf{d}_j\mathbf{c}_j^H$ with $\mathbf{c}_j$ that is highly sparse (and non-zero), and the objective would be invariant[1] to (arbitrarily) large scalings of $\mathbf{c}_j$ (i.e., non-coercive objective). The ℓ_∞ constraints on the columns of **C** (that constrain the magnitudes of entries of **C**) alleviate possible problems (e.g., unbounded iterates in algorithms) due to such a non-coercive objective.

Problem (P1) aims to learn the factors $\{\mathbf{d}_j\}_{j=1}^J$ and $\{\mathbf{c}_j\}_{j=1}^J$ that enable the best SOUP sparse representation of **Y**. However, (P1), like (P0), is non-convex, even if one replaces the ℓ_0 "norm" with a convex penalty.

Unlike the sparsity constraints in (P0), the term $\|\mathbf{C}\|_0 = \sum_{j=1}^J \|\mathbf{c}_j\|_0 = \sum_{i=1}^N \|\mathbf{x}_i\|_0 = \|\mathbf{X}\|_0$ in Problem (P1) (or (2)) penalizes the number of non-zeros in the (entire) coefficient matrix (i.e., the number of non-zeros used to represent a collection of signals), allowing variable sparsity levels across the signals. This flexibility could enable better data representation error versus sparsity trade-offs than with a fixed column sparsity constraint (as in (P0)). For example, in imaging or image processing applications, the dictionary is usually learned for image patches. Patches from different regions of an image typically contain different amounts of information[2], and thus enforcing a common sparsity bound for various patches does not reflect typical image properties (i.e., is restrictive) and usually leads to sub-optimal performance in applications. In contrast, Problem (P1) encourages a more general and flexible image model, and leads to promising performance in the experiments of this work. Additionally, we have observed that the different columns of **C** (or rows of **X**) learned by the proposed algorithm (in Section III) for (P1) typically have widely different sparsity levels or number of non-zeros in practice.

B. Alternative Formulations

Several variants of Problem (P1) could be constructed that also involve the SOUP representation. For example, the ℓ_0 "norm" for sparsity could be replaced by the ℓ_1 norm [51] resulting in the following formulation:

$$\begin{aligned} \min_{\{\mathbf{d}_j,\mathbf{c}_j\}} \left\| \boldsymbol{Y} - \sum_{j=1}^J \mathbf{d}_j\mathbf{c}_j^H \right\|_F^2 + \mu \sum_{j=1}^J \|\mathbf{c}_j\|_1 \\ \text{s.t. } \|\mathbf{d}_j\|_2 = 1 \forall j. \end{aligned} \qquad \text{(P2)}$$

Here, $\mu > 0$, and the objective is coercive with respect to **C** because of the ℓ_1 penalty. Another alternative to (P1) enforces p-block-orthogonality constraints on **D**. The dictionary in this

[1] Such degenerate representations for Y, however, cannot be minimizers in the problem because they simply increase the ℓ_0 sparsity penalty without affecting the fitting error (the first term) in the cost.

[2] Here, the emphasis is on the required sparsity levels for encoding different patches. This is different from the motivation for multi-class models such as in [16], [56] (or [11], [18]), where patches from different regions of an image are assumed to contain different "types" of features or textures or edges, and thus common sub-dictionaries or sub-transforms are learned for groups of patches with similar features.

case is split into blocks (instead of individual atoms), each of which has *p* (unit norm) atoms that are orthogonal to each other. For *p* = 2, we would have (added) constraints such as $\mathbf{d}_{2j-1}^H \mathbf{d}_{2j}=0, 1 \leq j \leq J/2$. In the extreme (more constrained) case of *p* = *n*, the dictionary would be made of several square unitary[3] blocks (cf. [58]). For tensor-type data, (P1) can be modified by enforcing the dictionary atoms to be in the form of a Kronecker product. The algorithm proposed in Section III can be easily extended to accommodate several such variants of Problem (P1). We do not explore all such alternatives in this work due to space constraints, and a more detailed investigation of these is left for future work.

III. Learning Algorithm and Properties

A. Algorithm

We apply a block coordinate descent method to estimate the unknown variables in Problem (P1). For each *j* (1 ≤ *j* ≤ *J*), the algorithm has two steps. First, we solve (P1) with respect to $\mathbf{c}_j$ keeping all the other variables fixed. We refer to this step as the *sparse coding step* in our method. Once $\mathbf{c}_j$ is updated, we solve (P1) with respect to $\mathbf{d}_j$ keeping all other variables fixed. This step is referred to as the *dictionary atom update step* or simply *dictionary update step*. The algorithm thus updates the factors of the various rank-one matrices one-by-one. The approach for (P2) is similar and is a simple extension of the OS-DL method in [51] to the complex-valued setting. We next describe the sparse coding and dictionary atom update steps of the methods for (P1) and (P2).

1) Sparse Coding Step for (P1)—Minimizing (P1) with respect to $\mathbf{c}_j$ leads to the following non-convex problem, where $\mathbf{E}_j \triangleq \boldsymbol{Y} - \sum_{k \neq j} \mathbf{d}_k \mathbf{c}_k^H$ is a fixed matrix based on the most recent values of all other atoms and coefficients:

$$\min_{\mathbf{c}_j \in \mathbb{C}^N} \left\| \mathbf{E}_j - \mathbf{d}_j \mathbf{c}_j^H \right\|_F^2 + \lambda^2 \|\mathbf{c}_j\|_0 \text{ s.t. } \|\mathbf{c}_j\|_\infty \leq L. \quad (3)$$

The following proposition provides the solution to Problem (3), where the hard-thresholding operator $H_\lambda(\cdot)$ is defined as

$$(H_\lambda(\mathbf{b}))_i = \begin{cases} 0, & |b_i| < \lambda \\ b_i, & |b_i| \geq \lambda \end{cases} \quad (4)$$

with $\mathbf{b} \in \mathbb{C}^N$, and the subscript *i* above indexes vector entries. We use *b*i (without bold font) to denote the *i*th (scalar) element of a vector **b**. We assume that the bound $L > \lambda$ holds and let $\mathbf{1}_N$ denote a vector of ones of length *N*. The operation "⊙" denotes element-wise multiplication, and **z** = min(**a**, **u**) for vectors $\mathbf{a}, \mathbf{u} \in \mathbb{R}^N$ denotes the element-wise minimum

[3] Recent works have shown the promise of learned orthonormal (or unitary) dictionaries or sparsifying transforms in applications such as image denoising [17], [57]. Learned multi-class unitary models have been shown to work well in inverse problem settings such as in MRI [18], [56].

operation, i.e., $z_i = \min(a_i; b_i)$, $1 \leq i \leq N$. For a vector $\mathbf{c} \in \mathbb{C}^N$, $e^{j\angle \mathbf{c}} \in \mathbb{C}^N$ is computed element-wise, with "∠" denoting the phase.

Proposition 1: Given $\mathbf{E}_j \in \mathbb{C}^{n\times N}$ and $\mathbf{d}_j \in \mathbb{C}^n$, and assuming $L > \lambda$, a global minimizer of the sparse coding problem (3) is obtained by the following truncated hard-thresholding operation:

$$\hat{\mathbf{c}}_j = \min\left(\left|H_\lambda\left(\mathbf{E}_j^H \mathbf{d}_j\right)\right|, L\mathbf{1}_N\right) \odot e^{j\angle \mathbf{E}_j^H \mathbf{d}_j}. \quad (5)$$

The minimizer of (3) is unique if and only if the vector $\mathbf{E}_j^H \mathbf{d}_j$ has no entry with a magnitude of λ.

The proof of Proposition 1 is provided in the supplementary material.

2) Sparse Coding Step for (P2)—The sparse coding step of (P2) involves solving the following problem:

$$\min_{\mathbf{c}_j \in \mathbb{C}^N} \left\|\mathbf{E}_j - \mathbf{d}_j \mathbf{c}_j^H\right\|_F^2 + \mu \|\mathbf{c}_j\|_1. \quad (6)$$

The solution is given by the following proposition (proof in the supplement), and was previously discussed in [51] for the case of real-valued data.

Proposition 2: Given $\mathbf{E}_j \in \mathbb{C}^{n\times N}$ and $\mathbf{d}_j \in \mathbb{C}^n$, the unique global minimizer of the sparse coding problem (6) is

$$\hat{\mathbf{c}}_j = \max\left(\left|\mathbf{E}_j^H \mathbf{d}_j\right| - \frac{\mu}{2}\mathbf{1}_N, 0\right) \odot e^{j\angle \mathbf{E}_j^H \mathbf{d}_j}. \quad (7)$$

3) Dictionary Atom Update Step—Minimizing (P1) or (P2) with respect to $\mathbf{d}_j$ leads to the following problem:

$$\min_{\mathbf{d}_j \in \mathbb{C}^n} \left\|\mathbf{E}_j - \mathbf{d}_j \mathbf{c}_j^H\right\|_F^2 \text{ s.t. } \|\mathbf{d}_j\|_2 = 1. \quad (8)$$

Proposition 3 provides the closed-form solution for (8). The solution takes the form given in [54]. We briefly derive the solution in the supplementary material considering issues such as uniqueness.

Proposition 3: Given $\mathbf{E}_j \in \mathbb{C}^{n\times N}$ and $\mathbf{c}_j \in \mathbb{C}^N$, a global minimizer of the dictionary atom update problem (8) is

$$\hat{\mathbf{d}}_j = \begin{cases} \frac{\mathbf{E}_j \mathbf{c}_j}{\|\mathbf{E}_j \mathbf{c}_j\|_2}, & \text{if } \mathbf{c}_j \neq 0 \\ \mathbf{v}, & \text{if } \mathbf{c}_j = 0 \end{cases} \quad (9)$$

where $\mathbf{v}$ can be any unit ℓ_2 norm vector (i.e., on the unit sphere). In particular, here, we set $\mathbf{v}$ to be the first column of the $n \times n$ identity matrix. The solution is unique if and only if $\mathbf{c}_j \neq 0$.

4) Overall Algorithms—Fig. 1 shows the Sum of OUter Products DIctionary Learning (SOUP-DIL) Algorithm for Problem (P1), dubbed SOUP-DILLO in this case, due to the ℓ_0 "norm". The algorithm needs initial estimates $\left\{\mathbf{d}_j^0, \mathbf{c}_j^0\right\}_{j=1}^J$ for the variables. For example, the initial sparse coefficients could be set to zero, and the initial dictionary could be a known analytical dictionary such as the overcomplete DCT [8]. When $\mathbf{c}_j^t = 0$, setting $\mathbf{d}_j^t$ to be the first column of the identity matrix in the algorithm could also be replaced with other (equivalent) settings such as $\mathbf{d}_j^t = \mathbf{d}_j^{t-1}$or setting $\mathbf{d}_j^t$ to a random unit norm vector. All of these settings have been observed to work well in practice. A random ordering of the atom/sparse coefficient updates in Fig. 1, i.e., random j sequence, also works in practice in place of cycling in the same order 1 through J every iteration. One could also alternate several times between the sparse coding and dictionary atom update steps for each j. However, this variation would increase computation.

The method for (P2) differs from SOUP-DILLO in the sparse coding step (Proposition 2). From prior work [51], we refer to this method (for (P2)) as OS-DL. We implement this method in a similar manner as in Fig. 1 (for complex-valued data); unlike OS-DL in [51], our implementation does not compute the matrix $\mathbf{E}_j$ for each j.

Finally, while we interleave the sparse coefficient ($\mathbf{c}_j$) and atom ($\mathbf{d}_j$) updates in Fig. 1, one could also cycle first through all the columns of $\mathbf{C}$ and then through the columns of $\mathbf{D}$ in the block coordinate descent (SOUP) methods. Such an approach was adopted recently in [59] for ℓ_1 penalized dictionary learning. We have observed similar performance with such an alternative update ordering strategy compared to an interleaved update order. Although the convergence results in Section V are for the ordering in Fig. 1, similar results can be shown to hold with alternative (deterministic) orderings in various settings.

B. Computational Cost Analysis

For each iteration t in Fig. 1, SOUP-DILLO involves J sparse code and dictionary atom updates. The sparse coding and atom update steps involve matrix-vector products for computing $\mathbf{b}^t$ and $\mathbf{h}^t$, respectively.

Memory Usage—An alternative approach to the one in Fig. 1 involves computing $\mathbf{E}_j^t = \boldsymbol{Y} - \sum_{k<j} \mathbf{d}_k^t \left(\mathbf{c}_k^t\right)^H - \sum_{k>j} \mathbf{d}_k^{t-1} \left(\mathbf{c}_k^{t-1}\right)^H$ (as in Propositions 1 and 3) directly at the beginning of each inner j iteration. This matrix could be updated sequentially and efficiently for each j by adding and subtracting appropriate sparse rank-one matrices, as done in OS-DL in [51] for the ℓ_1 case. However, this alternative approach requires storing and updating $\mathbf{E}_j^t \in \mathbb{C}^{n\times N}$, which is a large matrix for large N and n. The procedure in Fig. 1 avoids this overhead (similar to the Approximate K-SVD approach [35]), and is faster and saves memory usage.

Computational Cost—We now discuss the cost of each sparse coding and atom update step in the SOUP-DILLO method of Fig. 1 (a similar discussion holds for the method for (P2)). Consider the tth iteration and the jth inner iteration in Fig. 1, consisting of the update of the jth dictionary atom $\mathbf{d}_j$ and its corresponding sparse coefficients $\mathbf{c}_j$. As in Fig. 1, let $\mathbf{D} \in \mathbb{C}^{n\times J}$ be the dictionary whose columns are the current estimates of the atoms (at the start of the jth inner iteration), and let $\mathbf{C} \in \mathbb{C}^{N\times J}$ be the corresponding sparse coefficients matrix. (The index t on $\mathbf{D}$ and $\mathbf{C}$ is dropped to keep the notation simple.) Assume that the matrix $\mathbf{C}$ has αNn non-zeros, with $\alpha \ll 1$ typically. This translates to an average of αn non-zeros per row of $\mathbf{C}$ or $\alpha Nn/J$ non-zeros per column of $\mathbf{C}$. We refer to α as the sparsity factor of $\mathbf{C}$.

The sparse coding step involves computing the right hand side of (10). While computing $\mathbf{Y}^H \mathbf{d}_j^{t-1}$ requires Nn multiply-add[4] operations, computing $\mathbf{C}\mathbf{D}^H \mathbf{d}_j^{t-1}$ using matrix-vector products requires $Jn + \alpha Nn$ multiply-add operations. The remainder of the operations in (10) and (11) have $O(N)$ cost.

Next, when $\mathbf{c}_j^t \neq \mathbf{0}$, the dictionary atom update is as per (12) and (13). Since $\mathbf{c}_j^t$ is sparse with say r_j non-zeros, computing $\mathbf{Y}\,\mathbf{c}_j^t$ in (12) requires nr_j multiply-add operations, and computing $\mathbf{D}\mathbf{C}^H\mathbf{c}_j^t$ requires less than $Jn+\alpha Nn$ multiply-add operations. The cost of the remaining operations in (12) and (13) is negligible.

Thus, the net cost of the $J \geq n$ inner iterations in iteration t in Fig. 1 is dominated (for $N \gg J, n$) by $NJn+2\alpha_m NJn+\beta Nn^2$, where α_m is the maximum sparsity factor of the estimated $\mathbf{C}$'s during the inner iterations, and β is the sparsity factor of the estimated $\mathbf{C}$ at the end of iteration t. Thus, the cost per iteration of the block coordinate descent SOUP-DILLO Algorithm is about $(1 + \alpha')NJn$, with $\alpha' \ll 1$ typically. On the other hand, the proximal alternating algorithm proposed recently by Bao et al. for (P1) [41], [55] (Algorithm 2 in [55]) has a per-iteration cost of at least $2NJn+6\alpha NJn+4\alpha Nn^2$. This is clearly more computation[5] than SOUP-DILLO. The proximal methods [55] also involve more parameters than direct block coordinate descent schemes.

Assuming $J \propto n$, the cost per iteration of the SOUP-DILLO Algorithm scales as $O(Nn^2)$. This is lower than the per-iteration cost of learning an $n \times J$ synthesis dictionary $\mathbf{D}$ using K-SVD [30], which scales[6] (assuming that the synthesis sparsity level $s \propto n$ and $J \propto n$ in K-SVD) as $O(Nn^3)$. SOUP-DILLO converges in few iterations in practice (cf. supplement).

[4] In the case of complex-valued data, this would be the complex-valued multiply-accumulate (CMAC) operation (cf. [60]) that requires 4 real-valued multiplications and 4 real-valued additions.

[5] Bao et al. also proposed another proximal alternating scheme (Algorithm 3 in [55]) for discriminative incoherent dictionary learning. However, this method, when applied to (P1) (as a special case of discriminative incoherent learning), has been shown in [55] to be much slower than the proximal Algorithm 2 [55] for (P1).

[6] When $s \propto n$ and $J \propto n$, the per-iteration computational cost of the efficient implementation of K-SVD [54] also scales similarly as $O(Nn^3)$.

Therefore, the per-iteration computational advantages may also translate to net computational advantages in practice. This low cost could be particularly useful for big data applications, or higher dimensional (3D or 4D) applications.

IV. Dictionary-Blind Image Reconstruction

A. Problem Formulations

Here, we consider the application of sparsity penalized dictionary learning to inverse problems. In particular, we use the following ℓ_0 aggregate sparsity penalized dictionary learning regularizer that is based on (P1)

$$\zeta(\mathbf{y}) = \frac{1}{\nu} \min_{\mathbf{D},\mathbf{X}} \sum_{i=1}^{N} \|\mathbf{P}_i\mathbf{y} - \mathbf{D}\mathbf{x}_i\|_2^2 + \lambda^2 \|\mathbf{X}\|_0 \\ \text{s.t. } \|\mathbf{d}_j\|_2 = 1, \ \|\mathbf{x}_i\|_\infty \le L \forall i, j$$

in (1) to arrive at the following dictionary-blind image reconstruction problem:

$$\min_{\mathbf{y},\mathbf{D},\mathbf{X}} \nu \|\mathbf{A}\mathbf{y} - \mathbf{z}\|_2^2 + \sum_{i=1}^{N} \|\mathbf{P}_i\mathbf{y} - \mathbf{D}\mathbf{x}_i\|_2^2 + \lambda^2 \|\mathbf{X}\|_0 \\ \text{s.t. } \|\mathbf{d}_j\|_2 = 1, \ \|\mathbf{x}_i\|_\infty \le L \forall i, j. \qquad \text{(P3)}$$

Here, $\mathbf{P}_i \in \mathbb{R}^{n \times p}$ is an operator that extracts a $\sqrt{n} \times \sqrt{n}$ patch (for a 2D image) of $\mathbf{y}$ as a vector $\mathbf{P}_i\mathbf{y}$, and $\mathbf{D} \in \mathbb{C}^{n \times J}$ is a (unknown) synthesis dictionary for the image patches. A total of N overlapping image patches are assumed, and $\nu > 0$ is a weight in (P3). We use $\mathbf{Y}$ to denote the matrix whose columns are the patches $\mathbf{P}_i\mathbf{y}$, and $\mathbf{X}$ (with columns $\mathbf{x}_i$) denotes the corresponding dictionary-sparse representation of $\mathbf{Y}$. All other notations are as before. Similarly as in (P1), we approximate the (unknown) patch matrix $\mathbf{Y}$ using a sum of outer products representation.

An alternative to Problem (P3) uses a regularizer $\zeta(\mathbf{y})$ based on Problem (P2) rather than (P1). In this case, we have the following ℓ_1 sparsity penalized dictionary-blind image reconstruction problem, where $\|\mathbf{X}\|_1 = \sum_{i=1}^{N} \|\mathbf{x}_i\|_1$:

$$\min_{\mathbf{y},\mathbf{D},\mathbf{X}} \nu \|\mathbf{A}\mathbf{y} - \mathbf{z}\|_2^2 + \sum_{i=1}^{N} \|\mathbf{P}_i\mathbf{y} - \mathbf{D}\mathbf{x}_i\|_2^2 + \mu \|\mathbf{X}\|_1 \\ \text{s.t. } \|\mathbf{d}_j\|_2 = 1 \forall j. \qquad \text{(P4)}$$

Similar to (P1) and (P2), the dictionary-blind image reconstruction problems (P3) and (P4) are non-convex. The goal in these problems is to learn a dictionary and sparse coefficients, and reconstruct the image using only the measurements $\mathbf{z}$.

B. Algorithms and Properties

We adopt iterative block coordinate descent methods for (P3) and (P4) that lead to highly efficient solutions for the corresponding subproblems. In the *dictionary learning step*, we minimize (P3) or (P4) with respect to (**D**, **X**) keeping **y** fixed. In the *image update step*, we solve (P3) or (P4) for the image **y** keeping the other variables fixed. We describe these steps below.

1) Dictionary Learning Step—Minimizing (P3) with respect to (**D**, **X**) involves the following problem:

$$\begin{aligned}&\min_{\mathbf{D},\mathbf{X}} \|\boldsymbol{Y}-\mathbf{D}\mathbf{X}\|_F^2+\lambda^2\|\mathbf{X}\|_0\\ &\text{s.t. } \|\mathbf{d}_j\|_2=1,\ \|\mathbf{x}_i\|_\infty \leq L \forall i,j.\end{aligned} \quad (14)$$

By using the substitutions **X** = $\mathbf{C}^H$ and $\mathbf{D}\mathbf{C}^H=\sum_{j=1}^{J}\mathbf{d}_j\mathbf{c}_j^H$, Problem (14) becomes (P1) [7]. We then apply the SOUP-DILLO algorithm in Fig. 1 to update the dictionary **D** and sparse coefficients **C**. In the case of (P4), when minimizing with respect to (**D**, **X**), we again set **X** = $\mathbf{C}^H$ and use the SOUP representation to recast the resulting problem in the form of (P2). The dictionary and coefficients are then updated using the OS-DL method.

2) Image Update Step—Minimizing (P3) or (P4) with respect to **y** involves the following optimization problem:

$$\min_{\mathbf{y}} \nu\|\mathbf{A}\mathbf{y}-\mathbf{z}\|_2^2+\sum_{i=1}^{N}\|\mathbf{P}_i\mathbf{y}-\mathbf{D}\mathbf{x}_i\|_2^2 \quad (15)$$

This is a least squares problem whose solution satisfies the following normal equation:

$$\left(\sum_{i=1}^{N}\mathbf{P}_i^T\mathbf{P}_i+\nu\mathbf{A}^H\mathbf{A}\right)\mathbf{y}=\sum_{i=1}^{N}\mathbf{P}_i^T\mathbf{D}\mathbf{x}_i+\nu\mathbf{A}^H\mathbf{z} \quad (16)$$

When periodically positioned, overlapping patches (patch overlap stride [13] denoted by *r*) are used, and the patches that overlap the image boundaries 'wrap around' on the opposite side of the image [13], then $\sum_{i=1}^{N}\mathbf{P}_i^T\mathbf{P}_i$ is a diagonal matrix. Moreover, when the patch stride $r = 1$, $\sum_{i=1}^{N}\mathbf{P}_i^T\mathbf{P}_i=\beta I$, with $\beta = n$. In general, the unique solution to (16) can be found using techniques such as conjugate gradients (CG). In several applications, the matrix $\mathbf{A}^H\mathbf{A}$ in (16) is diagonal (e.g., in denoising or in inpainting) or readily diagonalizable. In such cases, the solution to (16) can be found efficiently [8], [13]. Here, we consider single coil compressed sensing MRI [6], where $\mathbf{A} = \mathbf{F}_u \in \mathbb{C}^{m\times p}$ ($m \ll p$), the undersampled Fourier

[7] The ℓ_0 constraints on the columns of X translate to identical constraints on the columns of C.

encoding matrix. Here, the measurements **z** are samples in Fourier space (or k-space) of an object **y**, and we assume for simplicity that **z** is obtained by subsampling on a uniform Cartesian (k-space) grid. Denoting by $\mathbf{F} \in \mathbb{C}^{p \times p}$ the full Fourier encoding matrix with $\mathbf{F}^H\mathbf{F} = \mathbf{I}$ (normalized), we get $\mathbf{F}\mathbf{F}_{\mathrm{u}}^H\mathbf{F}_{\mathrm{u}}\mathbf{F}^H$ is a diagonal matrix of ones and zeros, with ones at entries correspond to sampled k-space locations. Using this in (16) yields the following solution in Fourier space [13] with $\mathbf{S} \triangleq \mathbf{F}\sum_{i=1}^{N}\mathbf{P}_i^T\mathbf{D}\mathbf{x}_i$, $\mathbf{S}_0 \triangleq \mathbf{F}\mathbf{F}_{\mathrm{u}}^H\mathbf{z}$, and $\beta = n$ (i.e., assuming $r = 1$):

$$\mathbf{F}\mathbf{y}\left(k_1, k_2\right) = \begin{cases} \frac{\mathbf{S}(k_1,k_2)}{\beta} & , \ (k_1, k_2) \notin \Omega \\ \frac{\mathbf{S}(k_1,k_2)+\nu\ \mathbf{S}_0(k_1,k_2)}{\beta+\nu} & , \ (k_1, k_2) \in \Omega \end{cases} \quad (17)$$

where (k_1, k_2) indexes k-space or frequency locations (2D coordinates), and Ω is the subset of k-space sampled. The **y** solving (16) is obtained by an inverse FFT of **Fy** in (17).

3) Overall Algorithms and Computational Costs—Fig. 2 shows the algorithms for (P3) and (P4), which we refer to as the SOUP-DILLO and SOUP-DILLI image reconstruction algorithms, respectively. The algorithms start with an initial $(\mathbf{y}^0, \mathbf{D}^0, \mathbf{X}^0)$ (e.g., $\mathbf{y}^0 = \mathbf{A}^{\dagger}\mathbf{z}$, and the other variables initialized as in Section III-A4). In applications such as inpainting or single coil MRI, the cost per outer (t) iteration of the algorithms is typically dominated by the dictionary learning step, for which (assuming $J \propto n$) the cost scales as $O(KNn^2)$, with K being the number of inner iterations of dictionary learning. On the other hand, recent image reconstruction methods involving K-SVD (e.g., DLMRI [13]) have a worse corresponding cost per outer iteration of $O(KNn^3)$.

V. Convergence Analysis

This section presents a convergence analysis of the algorithms for the non-convex Problems (P1)–(P4). Problem (P1) involves the non-convex ℓ_0 penalty for sparsity, the unit ℓ_2 norm constraints on atoms of **D**, and the term $\left\| \boldsymbol{Y} - \sum_{j=1}^{J}\mathbf{d}_j\mathbf{c}_j^H \right\|_F^2$ that is a non-convex function involving the products of multiple unknown vectors. The various algorithms discussed in Sections III and IV are exact block coordinate descent methods for (P1)-(P4). Due to the high degree of non-convexity involved, recent results on convergence of (exact) block coordinate descent methods [61] do not immediately apply (e.g., the assumptions in [61] such as block-wise quasiconvexity or other conditions do not hold here). More recent works [62] on the convergence of block coordinate descent schemes also use assumptions (such as multi-convexity, etc.) that do not hold here. While there have been recent works [63]–[67] studying the convergence of alternating proximal-type methods for non-convex problems, we focus on the exact block coordinate descent schemes of Sections III and IV due to their simplicity. We discuss the convergence of these algorithms to the critical points (or generalized stationary points [68]) in the problems. In the following, we present some definitions and notations, before stating the main results.

A. Definitions and Notations

A sequence $\{\mathbf{a}^t\} \subset \mathbb{C}^p$ has is an accumulation point **a**, if there a subsequence that converges to **a**. The constraints $\|\mathbf{d}_j\|_2 = 1$, $1 \le j \le J$, in (P1) can instead be added as penalties in the cost by using barrier functions $\chi(\mathbf{d}_j)$ (taking the value $+\infty$ when the norm constraint is violated, and zero otherwise). The constraints $\|\mathbf{c}_j\|_\infty \le L$, $1 \le j \le J$, (P1), can also in be similarly replaced with barrier penalties $\psi(\mathbf{c}_j)$ $\forall j$. Then, we rewrite (P1) in unconstrained form with the following objective:

$$f(\mathbf{C},\mathbf{D}) = f(\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_J,\mathbf{d}_1,\mathbf{d}_2,\ldots,\mathbf{d}_J) = \lambda^2 \sum_{j=1}^{J} \|\mathbf{c}_j\|_0 + \left\| \boldsymbol{Y} - \sum_{j=1}^{J} \mathbf{d}_j \mathbf{c}_j^H \right\|_F^2 + \sum_{j=1}^{J} \chi(\mathbf{d}_j) + \sum_{j=1}^{J} \psi(\mathbf{c}_j). \tag{18}$$

We rewrite (P2) similarly with an objective $\tilde{f}(\mathbf{C},\mathbf{D})$ obtained by replacing the ℓ_0 "norm" above with the ℓ_1 norm, and dropping the penalties $\psi(\mathbf{c}_j)$. We also rewrite (P3) and (P4) in terms of the variable $\mathbf{C} = \mathbf{X}^H$, and denote the corresponding unconstrained objectives (involving barrier functions) as $g(\mathbf{C}, \mathbf{D}, \mathbf{y})$ and $\tilde{g}(\mathbf{C},\mathbf{D},\mathbf{y})$, respectively.

The iterates computed in the tth outer iteration of SOUP-DILLO (or alternatively in OS-DL) are denoted by the pair of matrices $(\mathbf{C}^t, \mathbf{D}^t)$.

B. Results for (P1) and (P2)

First, we present a convergence result for the SOUP-DILLO algorithm for (P1) in Theorem 1. Assume that the initial $(\mathbf{C}^0, \mathbf{D}^0)$ satisfies the constraints in (P1).

Theorem 1—Let $\{\mathbf{C}^t, \mathbf{D}^t\}$ denote the bounded iterate sequence generated by the SOUP-DILLO Algorithm with data $\mathbf{Y} \in \mathbb{C}^{n \times N}$ and initial $(\mathbf{C}^0, \mathbf{D}^0)$. Then, the following results hold:

i. The objective sequence $\{f^t\}$ with $f^t \triangleq f(\mathbf{C}^t, \mathbf{D}^t)$ is monotone decreasing, and converges to a finite value, say $f^* = f^*(\mathbf{C}^0, \mathbf{D}^0)$.

ii. All the accumulation points of the iterate sequence are equivalent in the sense that they achieve the exact same value f^* of the objective.

iii. Suppose each accumulation point $(\mathbf{C}, \mathbf{D})$ of the iterate sequence is such that the matrix **B** with columns $\mathbf{b}_j = \mathbf{E}_j^H \mathbf{d}_j$ and $\mathbf{E}_j = \boldsymbol{Y} - \mathbf{D}\mathbf{C}^H + \mathbf{d}_j \mathbf{c}_j^H$, has no entry with magnitude λ. Then every accumulation point of the iterate sequence is a critical point of the objective $f(\mathbf{C}, \mathbf{D})$. Moreover, the two sequences with terms $\|\mathbf{D}^t - \mathbf{D}^{t-1}\|_F$ and $\|\mathbf{C}^t - \mathbf{C}^{t-1}\|_F$ respectively, both converge to zero.

Theorem 1 establishes that for an initial point the $(\mathbf{C}^0, \mathbf{D}^0)$ *bounded* iterate sequence in SOUP-DILLO is such that all its (compact set of) accumulation points achieve the same value f^* of the objective. They are equivalent in that sense. In other words, the iterate sequence converges to an equivalence class of accumulation points. The value of f^* could vary with different initalizations.

Theorem 1 (Statement (iii)) also establishes that every accumulation point of the iterates is a critical point of $f(\mathbf{C}, \mathbf{D})$, i.e., for each initial $(\mathbf{C}^0, \mathbf{D}^0)$, the iterate sequence converges to an equivalence class of critical points of f. The results $\|\mathbf{D}^t - \mathbf{D}^{t-1}\|_F \to 0$ and $\|\mathbf{C}^t - \mathbf{C}^{t-1}\|_F \to 0$ also imply that the sparse approximation to the data $\mathbf{Z}^t = \mathbf{D}^t(\mathbf{C}^t)^H$ satisfies $\|\mathbf{Z}^t - \mathbf{Z}^{t-1}\|_F \to 0$. These are necessary but not sufficient conditions for the convergence of the entire sequences $\{\mathbf{D}^t\}$, $\{\mathbf{C}^t\}$, and $\{\mathbf{Z}^t\}$. The assumption on the entries of the matrix $\mathbf{B}$ in Theorem 1 (i.e., $|b_{ji}| \neq \lambda$) is equivalent to assuming that for every $1 \leq j \leq J$, there is a unique minimizer of f with respect to $\mathbf{c}_j$ with all other variables fixed to their values in the accumulation point $(\mathbf{C}, \mathbf{D})$.

Although Theorem 1 uses a uniqueness condition with respect to each accumulation point (for Statement (iii)), the following conjecture postulates that provided the following Assumption 1 (that uses a probabilistic model for the data) holds, the uniqueness condition holds with probability 1, i.e., the probability of a tie in assigning sparse codes is zero.

Assumption 1: The signals $\mathbf{y}_i \in \mathbb{C}^n$ for $1 \leq i \leq N$, are drawn independently from an absolutely continuous probability measure over the ball $S \triangleq \{\mathbf{y} \in \mathbb{C}^n: \|\mathbf{y}\|_2 \leq \beta_0\}$ for some $\beta_0 > 0$

Conjecture 1—Let Assumption 1 hold. Then, with probability 1, every accumulation point $(\mathbf{C}, \mathbf{D})$ of the iterate sequence in the SOUP-DILLO Algorithm is such that for each $1 \leq j \leq J$, the minimizer of $f(\mathbf{c}_1, \ldots, \mathbf{c}_{j-1}, \tilde{\mathbf{c}}_j, \mathbf{c}_{j+1}, \ldots, \mathbf{c}_J, \mathbf{d}_1, \ldots, \mathbf{d}_J)$ with respect to $\tilde{\mathbf{c}}_j$ is unique.

If Conjecture 1 holds, then every accumulation point of the iterates in SOUP-DILLO is immediately a critical point of $f(\mathbf{C}, \mathbf{D})$ with probability 1.

We now briefly state the convergence result for the OS-DL method for (P2). The result is more of a special case of Theorem 1. Here, the iterate sequence for an initial $(\mathbf{C}^0, \mathbf{D}^0)$ converges directly (without additional conditions) to an equivalence class (i.e., corresponding to a common objective value $\tilde{f}* = \tilde{f}*(\mathbf{C}^0, \mathbf{D}^0)$) of critical points of the objective $\tilde{f}(\mathbf{C}, \mathbf{D})$.

Theorem 2—Let $\{\mathbf{C}^t, \mathbf{D}^t\}$ denote the bounded iterate sequence generated by the OS-DL Algorithm with data $\mathbf{Y} \in \mathbb{C}^{n \times N}$ and initial $(\mathbf{C}^0, \mathbf{D}^0)$. Then, the iterate sequence converges to an equivalence class of critical points of $\tilde{f}(\mathbf{C}, \mathbf{D})$, and $\|\mathbf{D}^t - \mathbf{D}^{t-1}\|_F \to 0$ and $\|\mathbf{C}^t - \mathbf{C}^{t-1}\|_F \to 0$ as $t \to \infty$.

A brief proof of Theorem 1 is provided in the supplementary material. The proof for Theorem 2 is similar, as discussed in the supplement.

C. Results for (P3) and (P4)

First, we present the result for the SOUP-DILLO image reconstruction algorithm for (P3) in Theorem 3. We again assume that the initial $(\mathbf{C}^0, \mathbf{D}^0)$ satisfies the constraints in the problem. Recall that $\mathbf{Y}$ denotes the matrix with patches $\mathbf{P}_i\mathbf{y}$ for $1 \leq i \leq N$, as its columns.

Theorem 3—Let $\{\mathbf{C}^t, \mathbf{D}^t, \mathbf{y}^t\}$ denote the iterate sequence generated by the SOUP-DILLO image reconstruction Algorithm with measurements $\mathbf{z} \in \mathbb{C}^m$ and initial $(\mathbf{C}^0, \mathbf{D}^0, \mathbf{y}^0)$. Then, the following results hold:

i. The objective sequence $\{g^t\}$ with $g^t \triangleq g(\mathbf{C}^t, \mathbf{D}^t, \mathbf{y}^t)$ is monotone decreasing, and converges to a finite value, say $g^* = g^*(\mathbf{C}^0, \mathbf{D}^0, \mathbf{y}^0)$.

ii. The iterate sequence is bounded, and all its accumulation points are equivalent in the sense that they achieve the exact same value g^* of the objective.

iii. Each accumulation point $(\mathbf{C}, \mathbf{D}, \mathbf{y})$ of the iterate sequence satisfies

$$\mathbf{y} \in arg \min_{\tilde{\mathbf{y}}} g(\mathbf{C}, \mathbf{D}, \tilde{\mathbf{y}}) \qquad (19)$$

iv. As $t \to \infty$, $\| \mathbf{y}^t - \mathbf{y}^{t-1} \|_2$ converges to zero.

v. Suppose each accumulation point $(\mathbf{C}, \mathbf{D}, \mathbf{y})$ of the iterates is such that the matrix $\mathbf{B}$ with columns $\mathbf{b}_j = \mathbf{E}_j^H \mathbf{d}_j$ and $\mathbf{E}_j = Y - \mathbf{D}\mathbf{C}^H + \mathbf{d}_j \mathbf{c}_j^H$, has no entry with magnitude λ. Then every accumulation point of the iterate sequence is a critical point of the objective g. Moreover, $\| \mathbf{D}^t - \mathbf{D}^{t-1} \|_F \to 0$ and $\| \mathbf{C}^t - \mathbf{C}^{t-1} \|_F \to 0$ as $t \to \infty$.

Statements (i) and (ii) of Theorem 3 establish that for each initial $(\mathbf{C}^0, \mathbf{D}^0, \mathbf{y}^0)$, the bounded iterate sequence in the SOUP-DILLO image reconstruction algorithm converges to an equivalence class (common objective value) of accumulation points. Statements (iii) and (iv) establish that each accumulation point is a partial global minimizer (i.e., minimizer with respect to some variables while the rest are kept fixed) of $g(\mathbf{C}, \mathbf{D}, \mathbf{y})$ with respect to $\mathbf{y}$, and that $\| \mathbf{y}^t - \mathbf{y}^{t-1} \|_2 \to 0$ Statement (v) shows that the iterates converge to the critical points of g In fact, the accumulation points of the iterates can be shown to be partial global minimizers of $g(\mathbf{C}, \mathbf{D}, \mathbf{y})$ with respect to each column of $\mathbf{C}$ or $\mathbf{D}$. Statement (v) also establishes the properties $\| \mathbf{D}^t - \mathbf{D}^{t-1} \|_F \to 0$ and $\| \mathbf{C}^t - \mathbf{C}^{t-1} \|_F \to 0$. Similarly as in Theorem 1, Statement (v) of Theorem 3 uses a uniqueness condition with respect to the accumulation points of the iterates.

Finally, we briefly state the convergence result for the SOUP-DILLI image reconstruction Algorithm for (P4). The result is a special version of Theorem 3, where the iterate sequence for an initial $(\mathbf{C}^0, \mathbf{D}^0, \mathbf{y}^0)$ converges directly (without additional conditions) to an equivalence class (i.e., corresponding to a common objective value $\tilde{g}* = \tilde{g}*(\mathbf{C}^0, \mathbf{D}^0)$) of critical points of the objective $\tilde{g}$.

Theorem 4—Let $\{\mathbf{C}^t, \mathbf{D}^t, \mathbf{y}^t\}$ denote the iterate sequence generated by the SOUP-DILLI image reconstruction Algorithm for (P4) with measurements $\mathbf{z} \in \mathbb{C}^m$ and initial $(\mathbf{C}^0, \mathbf{D}^0, \mathbf{y}^0)$. Then, the iterate sequence converges to an equivalence class of critical points of $\tilde{g}(\mathbf{C}, \mathbf{D}, \mathbf{y})$. Moreover, $\| \mathbf{D}^t - \mathbf{D}^{t-1} \|_F \to 0$ and $\| \mathbf{C}^t - \mathbf{C}^{t-1} \|_F \to 0$, and $\| \mathbf{y}^t - \mathbf{y}^{t-1} \|_2 \to 0$ as $t \to \infty$.

A brief proof sketch for Theorems 3 and 4 is provided in the supplementary material.

The convergence results for the algorithms in Figs. 1 and 2 use the deterministic and cyclic ordering of the various updates (of variables). Whether one could generalize the results to other update orders (such as stochastic) is an interesting question that we leave for future work.

VI. Numerical Experiments

A. Framework

This section presents numerical results illustrating the convergence behavior as well as the usefulness of the proposed methods in applications such as sparse data representation and inverse problems. An empirical convergence study of the dictionary learning methods is included in the supplement. We used a large $L = 10^8$ in all experiments and the ℓ_∞ constraints were never active.

Section VI-B illustrates the quality of sparse data representations obtained using the SOUP-DILLO method, where we consider data formed using vectorized 2D patches of natural images. We compare the sparse representation quality obtained with SOUP-DILLO to that obtained with OS-DL (for (P2)), and the recent proximal alternating dictionary learning (which we refer to as PADL) algorithm for (P1) [41], [55] (Algorithm 2 in [55]). We used the publicly available implementation of the PADL method [73], and implemented OS-DL in a similar (memory efficient) manner as in Fig. 1. We measure the quality of trained sparse representation of data **Y** using the *normalized sparse representation error* (NSRE) $\|\mathbf{Y} - \mathbf{DC}^H\|_F / \|\mathbf{Y}\|_F$.

Results obtained using the SOUP-DILLO (learning) algorithm for image denoising are reported in [74]. We have briefly discussed these results in Appendix A for completeness. In the experiments of this work, we focus on general inverse problems involving non-trivial sensing matrices **A**, where we use the iterative dictionary-blind image reconstruction algorithms discussed in Section IV. In particular, we consider blind compressed sensing MRI [13], where $\mathbf{A} = \mathbf{F}_u$, the undersampled Fourier encoding matrix. Sections VI-C and VI-D examine the empirical convergence behavior and usefulness of the SOUP-DILLO and SOUP-DILLI image reconstruction algorithms for Problems (P3) and (P4), for blind compressed sensing MRI. We refer to our algorithms for (P3) and (P4) for (dictionary-blind) MRI as SOUP-DILLO MRI and SOUP-DILLI MRI, respectively. Unlike recent synthesis dictionary learning-based works [13], [50] that involve computationally expensive algorithms with no convergence analysis, our algorithms for (P3) and (P4) are efficient and have proven convergence guarantees.

Figure 3 shows the data (normalized to have unit peak pixel intensity) used in Sections VI-C and VI-D. In our experiments, we simulate undersampling of k-space with variable density 2D random sampling (feasible when data corresponding to multiple slices are jointly acquired, and the readout direction is perpendicular to image plane) [13], or using Cartesian sampling with variable density random phase encodes (1D random). We compare the reconstructions from undersampled measurements provided by SOUP-DILLO MRI and SOUP-DILLI MRI to those provided by the benchmark DLMRI method [13] that learns adaptive overcomplete dictionaries using K-SVD in a dictionary-blind image reconstruction framework. We also compare to the non-adaptive Sparse MRI method [6] that uses wavelets and total variation sparsity, the PANO method [75] that exploits the non-local similarities between image patches, and the very recent FDLCP method [18] that uses learned multi-class unitary dictionaries. Similar to prior work [13], we employ the peak-signal-to-noise ratio (PSNR) to measure the quality of MR image reconstructions. The PSNR (expressed in decibels (dB)) is computed as the ratio of the peak intensity value of a reference image to the root mean square reconstruction error (computed between image magnitudes) relative to the reference.

All our algorithm implementations were coded in Matlab R2015a. The computations in Section VI-B were performed with an Intel Xeon CPU X3230 at 2.66 GHz and 8 GB memory, employing a 64-bit Windows 7 operating system. The computations in Sections VI-C and VI-D were performed with an Intel Core i7 CPU at 2.6 GHz and 8 GB memory, employing a 64-bit Windows 7 operating system. A link to software to reproduce results in this work will be provided at http://web.eecs.umich.edu/~fessler/.

B. Adaptive Sparse Representation of Data

Here, we extracted 3×10^4 patches of size 8 × 8 from randomly chosen locations in the 512 × 512 standard images Barbara, Boats, and Hill. For this data, we learned dictionaries of size 64×256 for various choices of the parameter λ in (P1) (i.e., corresponding to a variety of solution sparsity levels). The initial estimate for **C** in SOUP-DILLO is an all-zero matrix, and the initial estimate for **D** is the overcomplete DCT [8], [76]. We measure the quality (performance) of adaptive data approximations $\mathbf{DC}^H$ using the NSRE metric. We also learned dictionaries using the recent methods for sparsity penalized dictionary learning in [41], [51]. All learning methods were initialized the same way. We are interested in the NSRE versus sparsity trade-offs achieved by different learning methods for the 3×10^4 image patches (rather than for separate test data)[8].

First, we compare the NSRE values achieved by SOUP-DILLO to those obtained using the recent PADL (for (P1)) approach [41], [55]. Both the SOUP-DILLO and PADL methods were simulated for 30 iterations for an identical set of λ values in (P1). We did not observe any marked improvements in performance with more iterations of learning. Since the PADL code [73] outputs only the learned dictionaries, we performed 60 iterations of block coordinate descent (over the $\mathbf{c}_j$'s in (P1)) to obtain the sparse coefficients with the learned dictionaries. Figs. 4(a) and 4(b) show the NSREs and sparsity factors obtained in SOUP-DILLO, and with learned PADL dictionaries for the image patch data. The proposed SOUP-DILLO achieves both lower NSRE (improvements up to 0.8 dB over the PADL dictionaries) and lower net sparsity factors. Moreover, it also has much lower learning times (Fig. 4(c)) than PADL.

Next, we compare the SOUP-DILLO and OS-DL methods for sparsely representing the same data. For completeness, we first show the NSRE versus sparsity trade-offs achieved

[8] This study is useful because in the dictionary-blind image reconstruction framework of this work, the dictionaries are adapted without utilizing separate training data. Methods that provide sparser adaptive representations of the underlying data also typically provide better image reconstructions in that setting [56].

during learning. Here, we measured the sparsity factors (of **C**) achieved within the schemes for various λ and μ values in (P1) and (P2), and then compared the NSRE values achieved within SOUP-DILLO and OS-DL at similar (achieved) sparsity factor settings. OS-DL ran for 30 iterations, which was sufficient for good performance. Fig. 4(e) shows the NSRE versus sparsity trade-offs achieved within the algorithms. SOUP-DILLO clearly achieves significantly lower NSRE values at similar net sparsities than OS-DL. Since these methods are for the ℓ_0 and ℓ_1 learning problems respectively, we also took the learned sparse coefficients in OS-DL in Fig. 4(e) and performed *debiasing* [77] by re-estimating the non-zero coefficient values (with supports fixed to the estimates in OS-DL) for each signal in a least squares sense to minimize the data fitting error. In this case, SOUP-DILLO in Fig. 4(e) provides an average NSRE improvement across various sparsities of 2.1 dB over OS-DL dictionaries. Since both SOUP-DILLO and OS-DL involve similar types of operations, their runtimes (Fig. 4(d)) for learning were quite similar. Next, when the dictionaries learned by OS-DL for various sparsities in Fig. 4(e) were used to estimate the sparse coefficients **C** in (P1) (using 60 iterations of ℓ_0 block coordinate descent over the $\mathbf{c}_j$'s and choosing the corresponding λ values in Fig. 4(e)); the resulting representations $\mathbf{DC}^H$ had on average worse NSREs and usually more nonzero coefficients than SOUP-DILLO. Fig. 4(f) plots the trade-offs. For example, SOUP-DILLO provides 3.15 dB better NSRE than the learned OS-DL dictionary (used with ℓ_0 sparse coding) at 7.5% sparsity. These results further illustrate the benefits of the learned models in SOUP-DILLO.

Finally, results included in the supplementary material show that when the learned dictionaries are used to sparse code (using orthogonal matching pursuit [22]) the data in a column-by-column (or signal-by-signal) manner, SOUP-DILLO dictionaries again outperform PADL dictionaries in terms of achieved NSRE. Moreover, at low sparsities, SOUP-DILLO dictionaries used with such column-wise sparse coding also outperformed (by 14–15 dB) dictionaries learned using K-SVD [30] (that is adapted for Problem (P0) with column-wise sparsity constraints). Importantly, at similar net sparsity factors, the NSRE values achieved by SOUP-DILLO in Fig. 4(e) tend to be quite a bit lower (better) than those obtained using the K-SVD method for (P0). Thus, solving Problem (P1) may offer potential benefits for adaptively representing data sets (e.g., patches of an image) using very few total non-zero coefficients. Further exploration of the proposed methods and comparisons for different dictionary sizes or larger datasets (of images or image patches) is left for future work.

C. Convergence of SOUP-DIL Image Reconstruction Algorithms in Dictionary-Blind Compressed Sensing MRI

Here, we consider the complex-valued reference image in Fig. 3(c) (Image (c)), and perform 2.5 fold undersampling of the k-space of the reference. Fig. 5(a) shows the variable density sampling mask. We study the behavior of the SOUP-DILLO MRI and SOUP-DILLI MRI algorithms for (P3) and (P4) respectively, when used to reconstruct the water phantom data from undersampled measurements. For SOUP-DILLO MRI, overlapping image patches of size 6 × 6 ($n = 36$) were used with stride $r = 1$ (with patch wrap around), $\nu = 10^6/p$ (with p the number of image pixels), and we learned a fourfold overcomplete (or 36 × 144) dictionary with $K = 1$ and $\lambda = 0.08$ in Fig. 2. The same settings were used for the SOUP-

DILLI MRI method for (P4) with $\mu = 0.08$. We initialized the algorithms with $\mathbf{y}^0 = \mathbf{A}^{\dagger}\mathbf{z}$, $\mathbf{C}^0 = \mathbf{0}$, and the initial $\mathbf{D}^0$ was formed by concatenating a square DCT dictionary with normalized random gaussian vectors.

Fig. 5 shows the behavior of the proposed dictionary-blind image reconstruction methods. The objective function values (Fig. 5(d)) in (P3) and (P4) decreased monotonically and quickly in the SOUP-DILLO MRI and SOUP-DILLI MRI algorithms, respectively. The initial reconstruction (Fig. 5(b)) shows large aliasing artifacts and has a low PSNR of 24.9 dB. The reconstruction PSNR (Fig. 5(e)), however, improves significantly over the iterations in the proposed methods and converges, with the final SOUP-DILLO MRI reconstruction (Fig. 5(c)) having a PSNR of 36.8 dB. For the ℓ_1 method, the PSNR converges to 36.4 dB, which is lower than for the ℓ_0 case. The sparsity factor for the learned coefficient matrix C was 5% for (P3) and 16% for (P4). Although larger values of μ decrease the sparsity factor for the learned C in (P4), we found that the PSNR also degrades for such settings in this example.

The changes between successive iterates $\|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2$ (Fig. 5(f)) or $\|\mathbf{C}^t - \mathbf{C}^{t-1}\|_F$ (Fig. 5(f)) or $\|\mathbf{D}^t - \mathbf{D}^{t-1}\|_F$ (Fig. 5(h)) decreased to small values for the proposed algorithms. Such behavior was predicted for the algorithms *by* Theorems 3 and 4, and is indicative (necessary but not suffficient condition) of convergence of the respective sequences.

Finally, Fig. 5 also shows the dictionary learned (jointly with the reconstruction) for image patches by SOUP-DILLO MRI along with the initial (Fig. 5(i)) dictionary. The learned synthesis dictionary is complex-valued whose real (Fig. 5(j)) and imaginary (Fig. 5(k)) parts are displayed, with the atoms shown as patches. The learned atoms appear quite different from the initial ones and display frequency or edge like structures that were learned efficiently from a few k-space measurements.

D. Dictionary-Blind Compressed Sensing MRI Results—We now consider images (a)-(g) in Fig. 3 and evaluate the efficacy of the proposed algorithms for (P3) and (P4) for reconstructing the images from undersampled k-space measurements. We compare the reconstructions obtained by the proposed methods to those obtained by the DLMRI [13], Sparse MRI [6], PANO [75], and FDLCP [18] methods. We used the built-in parameter settings in the publicly available implementations of Sparse MRI [78] and PANO [69], which performed well in our experiments. We used the zero-filling reconstruction as the initial guide image in PANO [69], [75].

We used the publicly available implementation of the multi-class dictionaries learning-based FDLCP method [79]. The ℓ_0 "norm"-based FDLCP was used in our experiments, as it was shown in [18] to outperform the ℓ_1 version. The built-in settings [79] for the FDLCP parameters such as patch size, λ, etc., performed well in our experiments, and we tuned the parameter β in each experiment to achieve the best image reconstruction quality.

For the DLMRI implementation [80], we used image patches of size[9] 6 × 6 [13], and learned a 36 × 144 dictionary and performed image reconstruction using 45 iterations of the algorithm. The patch stride $r = 1$, and 14400 randomly selected patches[10] were used during the dictionary learning step (executed with 20 iterations of K-SVD) of DLMRI. Mean-subtraction was not performed for the patches prior to the dictionary learning step. (We adopted this strategy for DLMRI here as it led to better performance.) A maximum sparsity level (of $s = 7$ per patch) is employed together with an error threshold (for sparse coding) during the dictionary learning step. The ℓ_2 error threshold per patch varies linearly from 0.34 to 0.04 over the DLMRI iterations, except for Figs. 3(a), 3(c), and 3(f) (noisier data), where it varies from 0.34 to 0.15 over the iterations. Once the dictionary is learnt in the dictionary learning step of each DLMRI (outer) iteration, all image patches are sparse coded with the same error threshold as used in learning and a relaxed maximum sparsity level of 14. This relaxed sparsity level is indicated in the DLMRI-Lab toolbox [80], as it leads to better performance in practice. As an example, DLMRI with these parameter settings provides 0.4 dB better reconstruction PSNR for the data in Fig. 5 compared to DLMRI with a common maximum sparsity level (other parameters as above) of $s = 7$ in the dictionary learning and follow-up sparse coding (of all patches) steps. We observed the above parameter settings (everything else as per the indications in the DLMRI-Lab toolbox [80]) to work well for DLMRI in the experiments.

For SOUP-DILLO MRI and SOUP-DILLI MRI, patches of size 6×6 were again used ($n = 36$ like for DLMRI) with stride $r = 1$ (with patch wrap around), $\nu = 10^6/p$, $M = 45$ (same number of outer iterations as for DLMRI), and a 36 × 144 dictionary was learned. We found that using larger values of λ or μ during the initial outer iterations of the methods led to faster convergence and better aliasing removal. Hence, we vary λ from 0.35 to 0.01 over the (outer t) iterations in Fig. 2, except for Figs. 3(a), 3(c), and 3(f) (noisier data), where it varies from 0.35 to 0.04. These settings and $\mu = \lambda/1.4$ worked well in our experiments. We used 5 inner iterations of SOUP-DILLO and 1 inner iteration (observed optimal) of OS-DL. The iterative reconstruction algorithms were initialized as mentioned in Section VI-C.

Table I lists the reconstruction PSNRs[11] corresponding to the zero-filling (the initial $\mathbf{y}^0$ in our methods), Sparse MRI, DLMRI, PANO, SOUP-DILLO MRI, and SOUP-DILLI MRI reconstructions for several cases. The proposed SOUP-DILLO MRI Algorithm for (P3) provides the best reconstruction PSNRs in Table I. In particular, it provides 1 dB better PSNR on the average compared to the K-SVD [30] based DLMRI method and the non-local patch similarity-based PANO method. While the K-SVD-based algorithm for image denoising [8] explicitly uses information of the noise variance (of Gaussian noise) in the observed noisy patches, in the compressed sensing MRI application here, the artifact properties (variance or distribution of the aliasing/noise artifacts) in each iteration of DLMRI are typically unknown, i.e., the DLMRI algorithm does not benefit from a superior modeling of artifact statistics and one must empirically set parameters such as the patch-wise error thresholds. The improvements provided by SOUP-DILLO MRI over DLMRI thus might stem from a better optimization framework for the former (e.g., the overall sparsity penalized formulation or the exact and guaranteed block coordinate descent algorithm). A

[9] The reconstruction quality improves slightly with a larger patch size, but with a substantial increase in runtime.

[10] Using a larger training size during the dictionary learning step of DLMRI provides negligible improvement in image reconstruction quality, while leading to increased runtimes. A different random subset is used in each iteration of DLMRI.

[11] While we compute PSNRs using magnitudes (typically the useful component of the reconstruction) of images, we have observed similar trends as in Table I when the PSNR is computed based on the difference (error) between the complex-valued images.

more detailed theoretical analysis including the investigation of plausible recovery guarantees for the proposed schemes is left for future work.

SOUP-DILLO MRI (average runtime of 2180 seconds) was also faster in Table I than the previous DLMRI (average runtime of 3156 seconds). Both the proposed SOUP methods significantly improved the reconstruction quality compared to the classical non-adaptive Sparse MRI method. Moreover, the ℓ_0 "norm"-based SOUP-DILLO MRI outperformed the corresponding ℓ_1 method (SOUP-DILLI MRI) by 1.4 dB on average in Table I, indicating potential benefits for ℓ_0 penalized dictionary adaptation in practice. The promise of non-convex sparsity regularizers (including the ℓ_0 or ℓ_p norm for $p < 1$) compared to ℓ_1 norm-based techniques for compressed sensing MRI has been demonstrated in prior works [18], [81], [82].

Table II compares the reconstruction PSNRs obtained by SOUP-DILLO MRI to those obtained by the recent ℓ_0 "norm"-based FDLCP [18] for the same cases as in Table I. SOUP-DILLO MRI initialized with zero-filling reconstructions performs quite similarly on the average (0.1 dB worse) as ℓ_0 FDLCP in Table II. However, with better initializations, SOUP-DILLO MRI can provide even better reconstructions than with the zero-filling initialization. We investigated SOUP-DILLO MRI, but initialized with the ℓ_0 FDLCP reconstructions (for **y**). The parameter λ was set to the eventual value in Table I, i.e., 0.01 or 0.04 (for noisier data), with decreasing λ's used for Image (c) with 2.5× Cartesian undersampling, where the FDLCP reconstruction was still highly aliased. In this case, SOUP-DILLO MRI consistently improved over the ℓ_0 FDLCP reconstructions (initializations), and provided 0.8 dB better PSNR on the average in Table II. These results illustrate the benefits and potential for the proposed dictionary-blind compressed sensing approaches. The PSNRs for our schemes could be further improved with better parameter selection strategies.

Fig. 6 shows the reconstructions and reconstruction error maps (i.e., the magnitude of the difference between the magnitudes of the reconstructed and reference images) for various methods for an example in Table I. The reconstructed images and error maps for SOUP-DILLO MRI show much fewer artifacts and smaller distortions than for the other methods. Another comparison is included in the supplement.

VII. Conclusions

This paper investigated in detail fast methods for synthesis dictionary learning. The SOUP algorithms for dictionary learning were further extended to the scenario of dictionary-blind image reconstruction. A convergence analysis was presented for the various efficient algorithms in highly non-convex problem settings. The proposed SOUP-DILLO algorithm for aggregate sparsity penalized dictionary learning had superior performance over recent dictionary learning methods for sparse data representation. The proposed SOUP-DILLO (dictionary-blind) image reconstruction method outperformed standard benchmarks involving the K-SVD algorithm, as well as some other recent methods in the compressed sensing MRI application. Recent works have investigated the data-driven adaptation of alternative signal models such as the analysis dictionary [14] or transform model [4], [15], [16], [56]. While we focused on synthesis dictionary learning methodologies in this work, we plan to compare various kinds of data-driven models in future work. We have considered extensions of the SOUP-DIL methodology to other novel settings and applications elsewhere [83]. Extensions of the SOUP-DIL methods for online learning [12] or for learning multi-class models are also of interest, and are left for future work.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

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 EB018753, and a UM-SJTU seed grant.

Appendix A Discussion of Image Denoising Results For SOUP-DILLO in [74]

Results obtained using the SOUP-DILLO (learning) algorithm for image denoising are reported in [74], where the results were compared to those obtained using the K-SVD image denoising method [8]. We briefly discuss these results here for completeness.

Recall that the goal in image denoising is to recover an estimate of an image $\mathbf{y} \in \mathbb{C}^p$ (2D image represented as a vector) from its corrupted measurements $\mathbf{z} = \mathbf{y} + \varepsilon$ where ε is the noise (e.g., i.i.d. Gaussian). First, while both K-SVD and the SOUP-DILLO (for (P1)) methods could be applied to noisy image patches to obtain adaptive denoising (as $\mathbf{Dx}_i$ in $\mathbf{P}_i\mathbf{z} \approx \mathbf{Dx}_i$) of the patches (the denoised image is obtained easily from denoised patches by averaging together the overlapping patches at their respective 2D locations, or solving (22) in [74]), the K-SVD-based denoising method [8] uses a dictionary learning procedure where the ℓ_0 "norms" of the sparse codes are minimized so that a fitting constraint or error constraint of $\|\mathbf{P}_i\mathbf{z} - \mathbf{Dx}_i\|_2^2 \leq \varepsilon$ is met for representing each noisy patch. In particular, when the noise is i.i.d. Gaussian, $\varepsilon = nC^2\sigma^2$ is used, with $C > 1$ (typically chosen very close to 1) a constant and σ^2 being the noise variance for pixels. Such a constraint serves as a strong prior (law of large numbers), and is an important reason for the denoising capability of K-SVD [8].

In the SOUP-DILLO denoising method in [74], we set $\lambda \propto \sigma$ during learning (in (P1)), and once the dictionary is learned from noisy image patches, we re-estimated the patch sparse codes using a single pass (over the noisy patches) of orthogonal matching pursuit (OMP) [22], by employing an error constraint criterion like in K-SVD denoising. This strategy only uses information on the Gaussian noise statistics in a sub-optimal way, especially during learning. However, SOUP-DILLO still provided comparable denoising performance visa-vis K-SVD with this approach (cf. [74]). Importantly, SOUP-DILLO provided up to 0.1–0.2 dB better denoising PSNR than K-SVD in (very) high noise cases in [74].

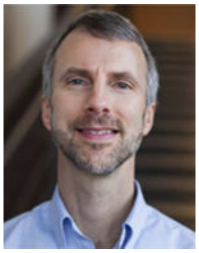
Biographies

Saiprasad Ravishankar received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Madras, in 2008. He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering, in 2010 and 2014 respectively, from the University of Illinois at Urbana-Champaign, where he was an Adjunct Lecturer in the Department of Electrical and Computer Engineering during Spring 2015, and a Postdoctoral Research Associate at the Coordinated Science Laboratory until August, 2015. Since then, he has been a Research Fellow in the Electrical Engineering and Computer Science Department at the University of Michigan. His current research interests include signal and image processing, signal modeling, data science, dictionary learning, biomedical and computational imaging, data-driven methods, inverse problems, compressed sensing, machine learning, and large-scale data processing. He was awarded the IEEE Signal Processing Society Young Author Best Paper Award for 2016.

Raj Rao Nadakuditi is an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He received his PhD in 2007 from the Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution. He was awarded an Office of Naval Research Young Investigator Award in 2011, an Air Force Office of Scientific Research Young Investigator Award in 2012, the Signal Processing Society Young Author Best Paper Award in 2012 and the DARPA Young Faculty Award in 2014. His research focuses on developing theory for random matrices for applications in signal processing, machine learning, queuing theory, and scattering theory.

Jeffrey A. Fessler is the William L. Root Professor of EECS at the University of Michigan. He received the BSEE degree from Purdue University in 1985, the MSEE degree from Stanford University in 1986, and the M.S. degree in Statistics from Stanford University in 1989. From 1985 to 1988 he was a National Science Foundation Graduate Fellow at Stanford, where he earned a Ph.D. in electrical engineering in 1990. He has worked at the University of Michigan since then. From 1991 to 1992 he was a Department of Energy

Alexander Hollaender Post-Doctoral Fellow in the Division of Nuclear Medicine. From 1993 to 1995 he was an Assistant Professor in Nuclear Medicine and the Bioengineering Program. He is now a Professor in the Departments of Electrical Engineering and Computer Science, Radiology, and Biomedical Engineering. He became a Fellow of the IEEE in 2006, for contributions to the theory and practice of image reconstruction. He received the Francois Erbsmann award for his IPMI93 presentation, and the Edward Hoffman Medical Imaging Scientist Award in 2013. He has served as an associate editor for IEEE Transactions on Medical Imaging, the IEEE Signal Processing Letters, and the IEEE Transactions on Image Processing, and is currently serving as an associate editor for the IEEE Transactions on Computational Imaging. He has chaired the IEEE T-MI Steering Committee and the ISBI Steering Committee. He was co-chair of the 1997 SPIE conference on Image Reconstruction and Restoration, technical program co-chair of the 2002 IEEE International Symposium on Biomedical Imaging (ISBI), and general chair of ISBI 2007. His research interests are in statistical aspects of imaging problems, and he has supervised doctoral research in PET, SPECT, X-ray CT, MRI, and optical imaging problems.



References

1. Elad M, Milanfar P, Rubinstein R. Analysis versus synthesis in signal priors. Inverse Problems. 2007; 23(3):947–968.
2. Candès EJ, Eldar YC, Needell D, Randall P. Compressed sensing with coherent and redundant dictionaries. Applied and Computational Harmonic Analysis. 2011; 31(1):59–73.
3. Pratt WK, Kane J, Andrews HC. Hadamard transform image coding. Proc IEEE. 1969; 57(1):58–68.
4. Ravishankar S, Bresler Y. Learning sparsifying transforms. IEEE Trans Signal Process. 2013; 61(5): 1072–1086.
5. Liu Y, Cai J-F, Zhan Z, Guo D, Ye J, Chen Z, Qu X. Balanced sparse model for tight frames in compressed sensing magnetic resonance imaging. PLOS ONE. Apr; 2015 10(4):1–19. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0119584.
6. Lustig M, Donoho D, Pauly J. Sparse MRI: The application of compressed sensing for rapid MR imaging. Magnetic Resonance in Medicine. 2007; 58(6):1182–1195. [PubMed: 17969013]
7. Liu Y, Zhan Z, Cai JF, Guo D, Chen Z, Qu X. Projected iterative soft-thresholding algorithm for tight frames in compressed sensing magnetic resonance imaging. IEEE Transactions on Medical Imaging. 2016; 35(9):2130–2140. [PubMed: 27071164]
8. Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans Image Process. 2006; 15(12):3736–3745. [PubMed: 17153947]
9. Mairal J, Elad M, Sapiro G. Sparse representation for color image restoration. IEEE Trans on Image Processing. 2008; 17(1):53–69.
10. Protter M, Elad M. Image sequence denoising via sparse and redundant representations. IEEE Trans on Image Processing. 2009; 18(1):27–36.
11. Ramirez I, Sprechmann P, Sapiro G. Classification and clustering via dictionary learning with structured incoherence and shared features. Proc IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2010. 2010:3501–3508.

12. Mairal J, Bach F, Ponce J, Sapiro G. Online learning for matrix factorization and sparse coding. J Mach Learn Res. 2010; 11:19–60.
13. Ravishankar S, Bresler Y. MR image reconstruction from highly undersampled k-space data by dictionary learning. IEEE Trans Med Imag. 2011; 30(5):1028–1041.
14. Rubinstein R, Peleg T, Elad M. Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model. IEEE Transactions on Signal Processing. 2013; 61(3):661–677.
15. Ravishankar S, Bresler Y. Learning doubly sparse transforms for images. IEEE Trans Image Process. 2013; 22(12):4598–4612. [PubMed: 23893720]
16. Wen B, Ravishankar S, Bresler Y. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. International Journal of Computer Vision. 2015; 114(2): 137–167.
17. Cai J-F, Ji H, Shen Z, Ye G-B. Data-driven tight frame construction and image denoising. Applied and Computational Harmonic Analysis. 2014; 37(1):89–105.
18. Zhan Z, Cai JF, Guo D, Liu Y, Chen Z, Qu X. Fast multiclass dictionaries learning with geometrical directions in mri reconstruction. IEEE Transactions on Biomedical Engineering. Sep; 2016 63(9):1850–1861. [PubMed: 26625404]
19. Vidal R. Subspace clustering. IEEE Signal Processing Magazine. 2011; 28(2):52–68.
20. Vidal, E Elhamifar R. Sparsity in unions of subspaces for classification and clustering of high-dimensional data. 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton). 2011:1085–1089.
21. Natarajan BK. Sparse approximate solutions to linear systems. SIAM J Comput. Apr.1995 24(2): 227–234.
22. Pati Y, Rezaiifar R, Krishnaprasad P. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. Asilomar Conf on Signals, Systems and Comput. 1993; 1:40–44.
23. Mallat SG, Zhang Z. Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing. 1993; 41(12):3397–3415.
24. Chen SS, Donoho DL, Saunders MA. Atomic decomposition by basis pursuit. SIAM J Sci Comput. 1998; 20(1):33–61.
25. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Annals of Statistics. 2004; 32:407–499.
26. Needell D, Tropp J. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis. 2009; 26(3):301–321.
27. Dai W, Milenkovic O. Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans Information Theory. 2009; 55(5):2230–2249.
28. Olshausen BA, Field DJ. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature. 1996; 381(6583):607–609. [PubMed: 8637596]
29. Engan K, Aase S, Hakon-Husoy J. Method of optimal directions for frame design. Proc IEEE International Conference on Acoustics, Speech, and Signal Processing. 1999:2443–2446.
30. Aharon M, Elad M, Bruckstein A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing. 2006; 54(11):4311–4322.
31. Yaghoobi M, Blumensath T, Davies M. Dictionary learning for sparse approximations with the majorization method. IEEE Transaction on Signal Processing. 2009; 57(6):2178–2191.
32. Kong S, Wang D. A dictionary learning approach for classification: Separating the particularity and the commonality. Proceedings of the 12th European Conference on Computer Vision. 2012:186–199.
33. Gribonval R, Schnass K. Dictionary identification–sparse matrix-factorization vial_1 -minimization. IEEE Trans Inform Theory. 2010; 56(7):3523–3539.
34. Barchiesi D, Plumbley MD. Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. IEEE Transactions on Signal Processing. 2013; 61(8):2055–2065.
35. Rubinstein R, Zibulevsky M, Elad M. Double sparsity: Learning sparse dictionaries for sparse signal approximation. IEEE Transactions on Signal Processing. 2010; 58(3):1553–1564.

36. Skretting K, Engan K. Recursive least squares dictionary learning algorithm. IEEE Transactions on Signal Processing. 2010; 58(4):2121–2130.

37. Ophir B, Lustig M, Elad M. Multi-scale dictionary learning using wavelets. IEEE Journal of Selected Topics in Signal Processing. 2011; 5(5):1014–1024.

38. Smith LN, Elad M. Improving dictionary learning: Multiple dictionary updates and coefficient reuse. IEEE Signal Processing Letters. Jan; 2013 20(1):79–82.

39. Sadeghi M, Babaie-Zadeh M, Jutten C. Dictionary learning for sparse representation: A novel approach. IEEE Signal Processing Letters. Dec; 2013 20(12):1195–1198.

40. Seghouane A-K, Hanif M. A sequential dictionary learning algorithm with enforced sparsity. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015:3876–3880.

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

42. Rakotomamonjy A. Direct optimization of the dictionary learning problem. IEEE Transactions on Signal Processing. 2013; 61(22):5495–5506.

43. Hawe S, Seibert M, Kleinsteuber M. Separable dictionary learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2013:438–445.

44. Spielman DA, Wang H, Wright J. Exact recovery of sparsely-used dictionaries. Proceedings of the 25th Annual Conference on Learning Theory. 2012:37.1–37.18.

45. Agarwal A, Anandkumar A, Jain P, Netrapalli P. Learning sparsely used overcomplete dictionaries via alternating minimization. SIAM Journal on Optimization. 2016; 26(4):2775–2799.

46. Arora S, Ge R, Moitra A. New algorithms for learning incoherent and overcomplete dictionaries. Proceedings of The 27th Conference on Learning Theory. 2014:779–806.

47. Xu Y, Yin W. A fast patch-dictionary method for whole image recovery. Inverse Problems and Imaging. 2016; 10(2):563–583.

48. Agarwal A, Anandkumar A, Jain P, Netrapalli P, Tandon R. Learning sparsely used overcomplete dictionaries. Journal of Machine Learning Research. 2014; 35:1–15.

49. Liao HY, Sapiro G. Sparse representations for limited data tomography. Proc IEEE International Symposium on Biomedical Imaging (ISBI). 2008:1375–1378.

50. Wang Y, Zhou Y, Ying L. Undersampled dynamic magnetic resonance imaging using patch-based spatiotemporal dictionaries. 2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI), April. 2013:294–297.

51. Sadeghi M, Babaie-Zadeh M, Jutten C. Learning overcomplete dictionaries based on atom-by-atom updating. IEEE Transactions on Signal Processing. 2014; 62(4):883–891.

52. Candès E, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans Information Theory. 2006; 52(2):489–509.

53. Donoho D. Compressed sensing. IEEE Trans Information Theory. 2006; 52(4):1289–1306.

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

55. Bao C, Ji H, Quan Y, Shen Z. Dictionary learning for sparse coding: Algorithms and convergence analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. Jul; 2016 38(7):1356–1369. [PubMed: 26452248]

56. Ravishankar S, Bresler Y. Data-driven learning of a union of sparsifying transforms model for blind compressed sensing. IEEE Transactions on Computational Imaging. 2016; 2(3):294–309.

57. Ravishankar S, Bresler Y. Closed-form solutions within sparsifying transform learning. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2013:5378–5382.

58. Lesage S, Gribonval R, Bimbot F, Benaroya L. Learning unions of orthonormal bases with thresholded singular value decomposition. Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing. 2005; 55:v/293–v/296.

59. Li Z, Ding S, Li Y. A fast algorithm for learning overcomplete dictionary for sparse representation based on proximal operators. Neural Computation. 2015; 27(9):1951–1982. [PubMed: 26161819]

60. Wefers, F. Partitioned convolution algorithms for real-time auralization. Berlin, Germany: Logos Verlag; 2015.

61. Tseng P. Convergence of a block coordinate descent method for nondifferentiable minimization. J Optim Theory Appl. 2001; 109(3):475–494.

62. Xu Y, Yin W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on Imaging Sciences. 2013; 6(3):1758–1789.

63. Attouch H, Bolte J, Redont P, Soubeyran A. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. Math Oper Res. May; 2010 35(2):438–457.

64. Bolte J, Sabach S, Teboulle M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math Program. 2014; 146(1):459–494.

65. Chouzenoux E, Pesquet J-C, Repetti A. A block coordinate variable metric forward-backward algorithm. Journal of Global Optimization. 2016; 66(3):457–485.

66. Abboud F, Chouzenoux E, Pesquet JC, Chenot JH, Laborelli L. A hybrid alternating proximal method for blind video restoration. 22nd European Signal Processing Conference (EUSIPCO). 2014:1811–1815.

67. Hesse R, Luke DR, Sabach S, Tam MK. Proximal heterogeneous block implicit-explicit method and application to blind ptychographic diffraction imaging. SIAM J Imaging Sciences. 2015; 8(1): 426–457.

68. Rockafellar, RT., Wets, RJB. Variational Analysis. Heidelberg, Germany: Springer-Verlag; 1998.

69. Qu, X. PANO Code. 2014. http://www.quxiaobo.org/project/CS_MRI_PANO/Demo_PANO_SparseMRI.zip[Online; accessed May 2015]

70. Qu, X. Water phantom. 2014. http://www.quxiaobo.org/project/MRI%20data/WaterPhantom.zip[Online; accessed September 2014]

71. Ye, JC. k-t FOCUSS software. 2012. http://bispl.weebly.com/k-t-focuss.html[Online; accessed July 2015]

72. Qu, X. Brain image. 2014. http://www.quxiaobo.org/project/MRI%20data/T2wBrain.zip[Online; accessed September 2014]

73. Bao, C., Ji, H., Quan, Y., Shen, Z. L0 norm based dictionary learning by proximal mathods. 2014. http://www.math.nus.edu.sg/~matjh/download/L0_dict_learning/L0_dict_learning_v1.1.zip[Online; accessed Apr 2016]

74. Ravishankar, S., Nadakuditi, RR., Fessler, JA. Efficient sum of outer products dictionary learning (SOUP-DIL) - the ℓ_0 method. 2015. http://arxiv.org/abs/1511.08842

75. Qu X, Hou Y, Lam F, Guo D, Zhong J, Chen Z. Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator. Medical Image Analysis. Aug; 2014 18(6):843–856. [PubMed: 24176973]

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

77. Figueiredo MAT, Nowak RD, Wright SJ. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE Journal of Selected Topics in Signal Processing. 2007; 1(4):586–597.

78. Lustig, M. Michael Lustig home page. 2014. http://www.eecs.berkeley.edu/~mlustig/Software.html[Online; accessed October 2014]

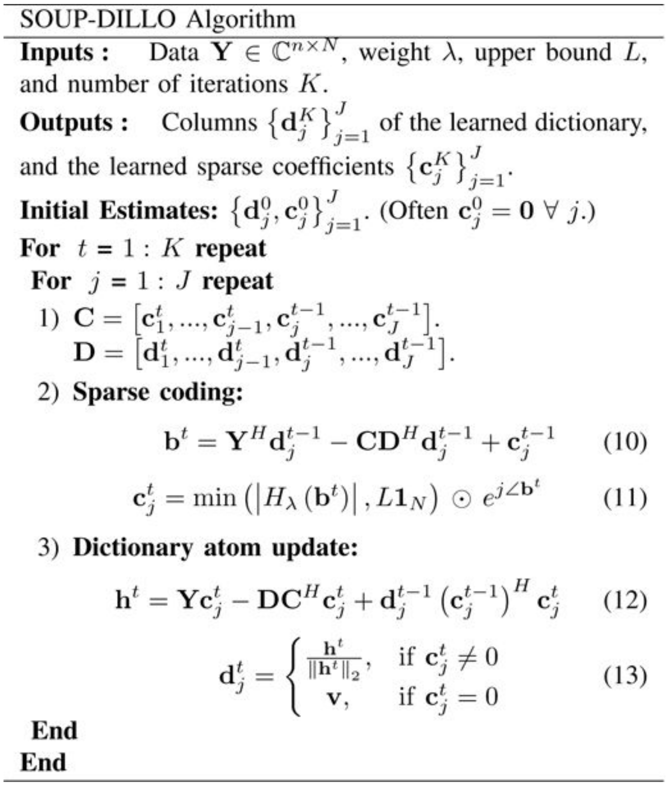
79. Zhan, Z., Qu, X. FDLCP Code. 2016. http://www.quxiaobo.org/project/CS_MRI_FDLCP/Demo_FDLCP_L1_L0.zip[Online; accessed January 2017]

80. Ravishankar, S., Bresler, Y. DLMRI - Lab: Dictionary learning MRI software. 2013. http://www.ifp.illinois.edu/~yoram/DLMRI-Lab/DLMRI.html[Online; accessed October 2014]

81. Trzasko J, Manduca A. Highly undersampled magnetic resonance image reconstruction via homotopicl_0-minimization. IEEE Trans Med Imaging. 2009; 28(1):106–121. [PubMed: 19116193]

82. Chartrand R. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. Proc IEEE International Symposium on Biomedical Imaging (ISBI). 2009:262–265.

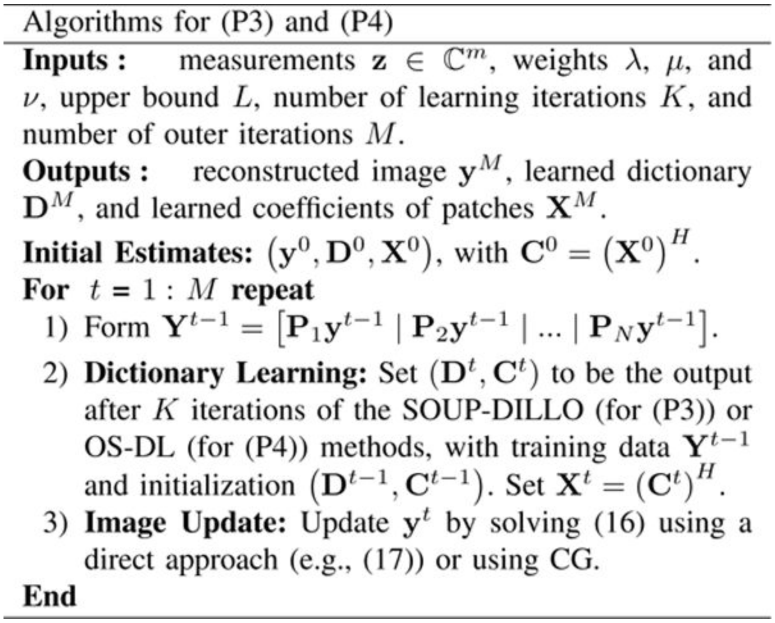
83. Ravishankar S, Moore BE, Nadakuditi RR, Fessler JA. LASSI: A low-rank and adaptive sparse signal model for highly accelerated dynamic imaging. IEEE Image Video and Multidimensional Signal Processing (IVMSP) workshop. 2016



```
SOUP-DILLO Algorithm
Inputs :  Data $\mathbf{Y} \in \mathbb{C}^{n \times N}$, weight $\lambda$, upper bound $L$, and number of iterations $K$.
Outputs :  Columns $\{\mathbf{d}_j^K\}_{j=1}^J$ of the learned dictionary, and the learned sparse coefficients $\{\mathbf{c}_j^K\}_{j=1}^J$.
Initial Estimates: $\{\mathbf{d}_j^0, \mathbf{c}_j^0\}_{j=1}^J$. (Often $\mathbf{c}_j^0 = \mathbf{0} \ \forall \ j$.)
For $t = 1 : K$ repeat
  For $j = 1 : J$ repeat
    1) $\mathbf{C} = [\mathbf{c}_1^t, ..., \mathbf{c}_{j-1}^t, \mathbf{c}_j^{t-1}, ..., \mathbf{c}_J^{t-1}]$.
       $\mathbf{D} = [\mathbf{d}_1^t, ..., \mathbf{d}_{j-1}^t, \mathbf{d}_j^{t-1}, ..., \mathbf{d}_J^{t-1}]$.
    2) Sparse coding:
       $\mathbf{b}^t = \mathbf{Y}^H \mathbf{d}_j^{t-1} - \mathbf{C}\mathbf{D}^H \mathbf{d}_j^{t-1} + \mathbf{c}_j^{t-1}$    (10)
       $\mathbf{c}_j^t = \min\left(\left|H_\lambda(\mathbf{b}^t)\right|, L\mathbf{1}_N\right) \odot e^{j\angle \mathbf{b}^t}$    (11)
    3) Dictionary atom update:
       $\mathbf{h}^t = \mathbf{Y}\mathbf{c}_j^t - \mathbf{D}\mathbf{C}^H \mathbf{c}_j^t + \mathbf{d}_j^{t-1}\left(\mathbf{c}_j^{t-1}\right)^H \mathbf{c}_j^t$    (12)
       $\mathbf{d}_j^t = \begin{cases} \frac{\mathbf{h}^t}{\|\mathbf{h}^t\|_2}, & \text{if } \mathbf{c}_j^t \neq 0 \\ \mathbf{v}, & \text{if } \mathbf{c}_j^t = 0 \end{cases}$    (13)
  End
End
```

Fig. 1.

The SOUP-DILLO Algorithm (due to the ℓ_0 "norm") for Problem (P1). Superscript t denotes the iterates in the algorithm. The vectors b^t and h^t above are computed efficiently via sparse operations.



```
Algorithms for (P3) and (P4)
Inputs : measurements $\mathbf{z} \in \mathbb{C}^m$, weights $\lambda$, $\mu$, and $\nu$, upper bound $L$, number of learning iterations $K$, and number of outer iterations $M$.
Outputs : reconstructed image $\mathbf{y}^M$, learned dictionary $\mathbf{D}^M$, and learned coefficients of patches $\mathbf{X}^M$.
Initial Estimates: $(\mathbf{y}^0, \mathbf{D}^0, \mathbf{X}^0)$, with $\mathbf{C}^0 = (\mathbf{X}^0)^H$.
For $t$ = $1 : M$ repeat
  1) Form $\mathbf{Y}^{t-1} = [\mathbf{P}_1\mathbf{y}^{t-1} \mid \mathbf{P}_2\mathbf{y}^{t-1} \mid \ldots \mid \mathbf{P}_N\mathbf{y}^{t-1}]$.
  2) Dictionary Learning: Set $(\mathbf{D}^t, \mathbf{C}^t)$ to be the output after $K$ iterations of the SOUP-DILLO (for (P3)) or OS-DL (for (P4)) methods, with training data $\mathbf{Y}^{t-1}$ and initialization $(\mathbf{D}^{t-1}, \mathbf{C}^{t-1})$. Set $\mathbf{X}^t = (\mathbf{C}^t)^H$.
  3) Image Update: Update $\mathbf{y}^t$ by solving (16) using a direct approach (e.g., (17)) or using CG.
End
```

Fig. 2.

The SOUP-DILLO and SOUP-DILLI image reconstruction algorithms for Problems (P3) and (P4), respectively. Superscript *t* denotes the iterates. Parameter *L* can be set very large in practice (e.g., $L \propto \|\mathbf{A}^{\dagger}\mathbf{z}\|_2$).

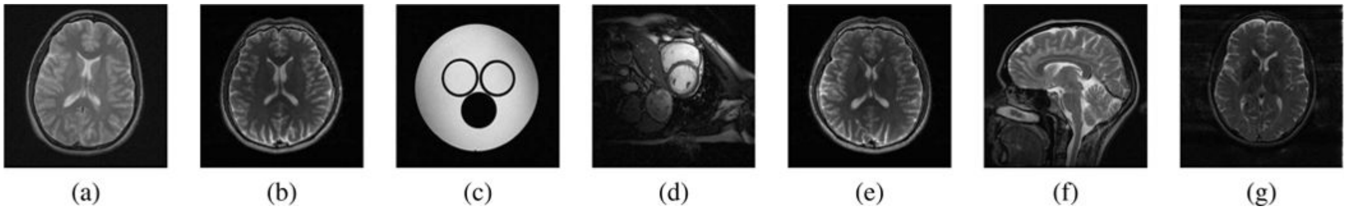
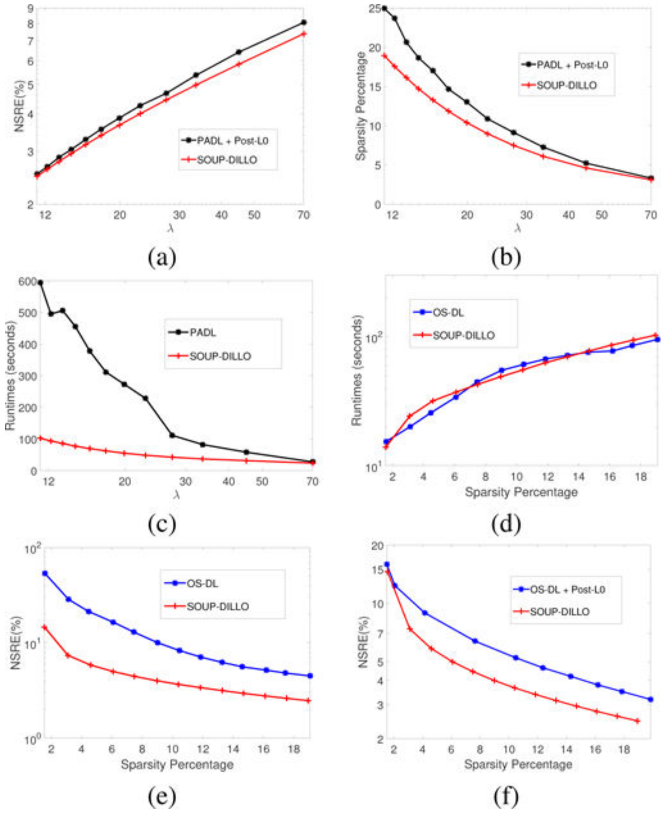


Fig. 3.

Test data (magnitudes of the complex-valued MR data are displayed here). Image (a) is available at http://web.stanford.edu/class/ee369c/data/brain.mat. The images (b)-(e) are publicly available: (b) T2 weighted brain image [69], (c) water phantom [70], (d) cardiac image [71], and (e) T2 weighted brain image [72]. Image (f) is a reference sagittal brain slice provided by Prof. Michael Lustig, UC Berkeley. Image (g) is a complex-valued reference SENSE reconstruction of 32 channel fully-sampled Cartesian axial data from a standard spin-echo sequence. Images (a) and (f) are 512 × 512, while the rest are 256 × 256. The images (b) and (g) have been rotated clockwise by 90°here for display purposes. In the experiments, we use the actual orientations.



(a) (b)

(c) (d)

(e) (f)

Fig. 4.

Comparison of dictionary learning approaches for adaptive sparse representation (NSRE and sparsity factors are expressed as percentages): (a) NSRE values for SOUP-DILLO at various λ along with those obtained by performing ℓ_0 block coordinate descent sparse coding (as in (P1)) using learned PADL [41], [55] dictionaries (denoted 'Post-L0' in the plot legend); (b) (net) sparsity factors for SOUP-DILLO at various λ along with those obtained by performing ℓ_0 block coordinate descent sparse coding using learned PADL [41], [55] dictionaries; (c) learning times for SOUP-DILLO and PADL; (d) learning times for SOUP-DILLO and OS-DL for various achieved (net) sparsity factors (in learning); (e) NSRE vs. (net) sparsity factors achieved within SOUP-DILLO and OS-DL; and (f) NSRE vs. (net) sparsity factors achieved within SOUP-DILLO along with those obtained by performing ℓ_0 (block coordinate descent) sparse coding using learned OS-DL dictionaries.

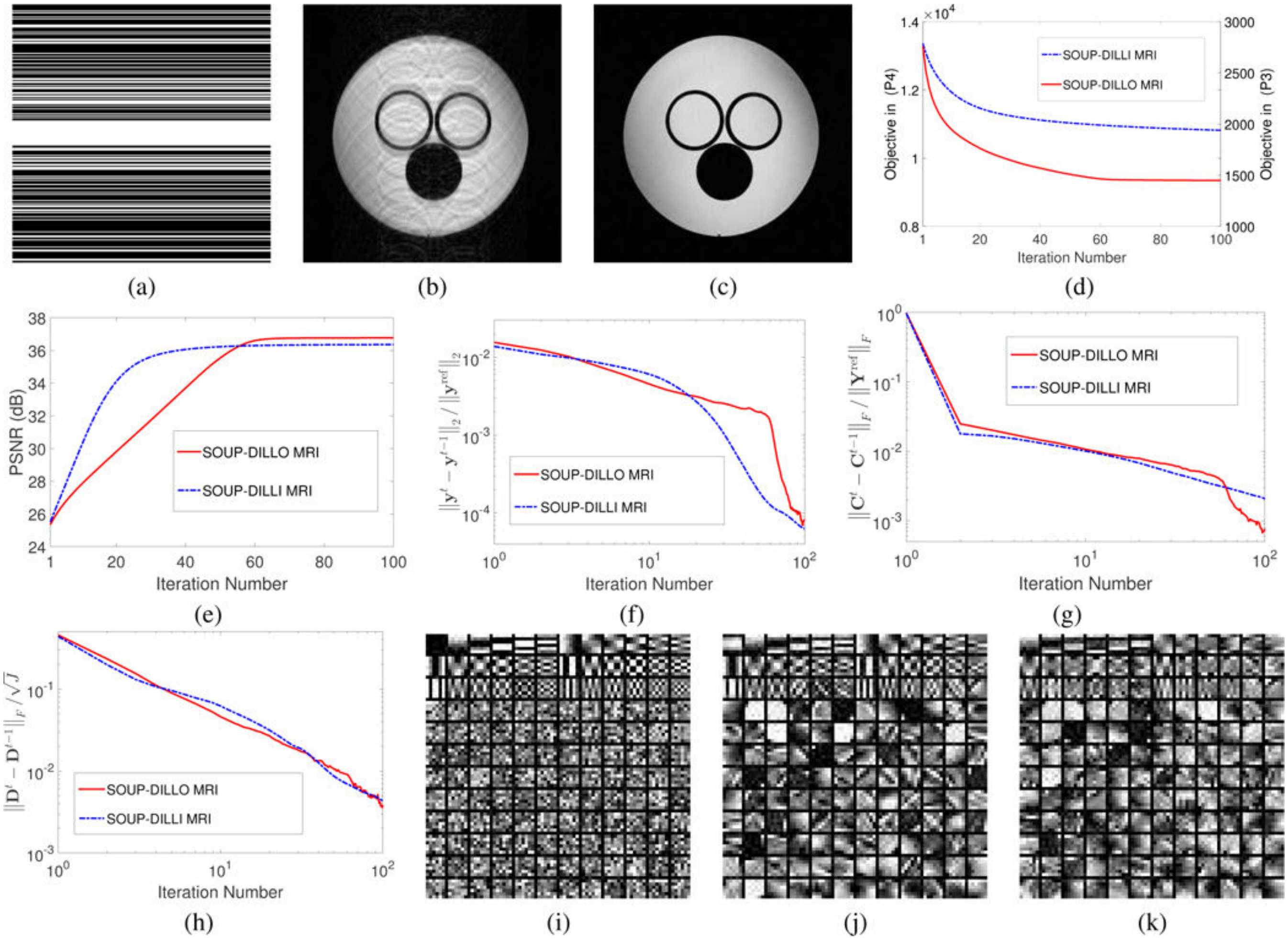


Fig. 5.

Behavior of SOUP-DILLO MRI (for (P3)) and SOUP-DILLI MRI (for (P4)) for Image (c) with Cartesian sampling and 2.5× undersampling: (a) sampling mask in k-space; (b) magnitude of initial reconstruction y^0 (PSNR = 24.9 dB); (c) SOUP DILLO MRI (final) reconstruction magnitude (PSNR = 36.8 dB); (d) objective function values for SOUP-DILLO MRI and SOUP-DILLI MRI; (e) reconstruction PSNR over iterations; (f) changes between successive image iterates ($\| \mathbf{y}^t - \mathbf{y}^{t-1} \|_2$) normalized by the norm of the reference image ($\| \mathbf{y}^{\text{ref}} \|_2 = 122.2$); (g) normalized changes between successive coefficient iterates ($\| \mathbf{C}^t - \mathbf{C}^{t-1} \|_F / \| \mathbf{Y}^{\text{ref}} \|_F$) where Y^{ref} is the patch matrix for the reference image; (h) normalized changes between successive dictionary iterates $(\|\mathbf{D}^t - \mathbf{D}^{t-1}\|_F / \sqrt{J})$ (i) initial real-valued dictionary in the algorithms; and (j) real and (k) imaginary parts of the learnt dictionary for SOUP-DILLO MRI. The dictionary columns or atoms are shown as 6 × 6 patches.

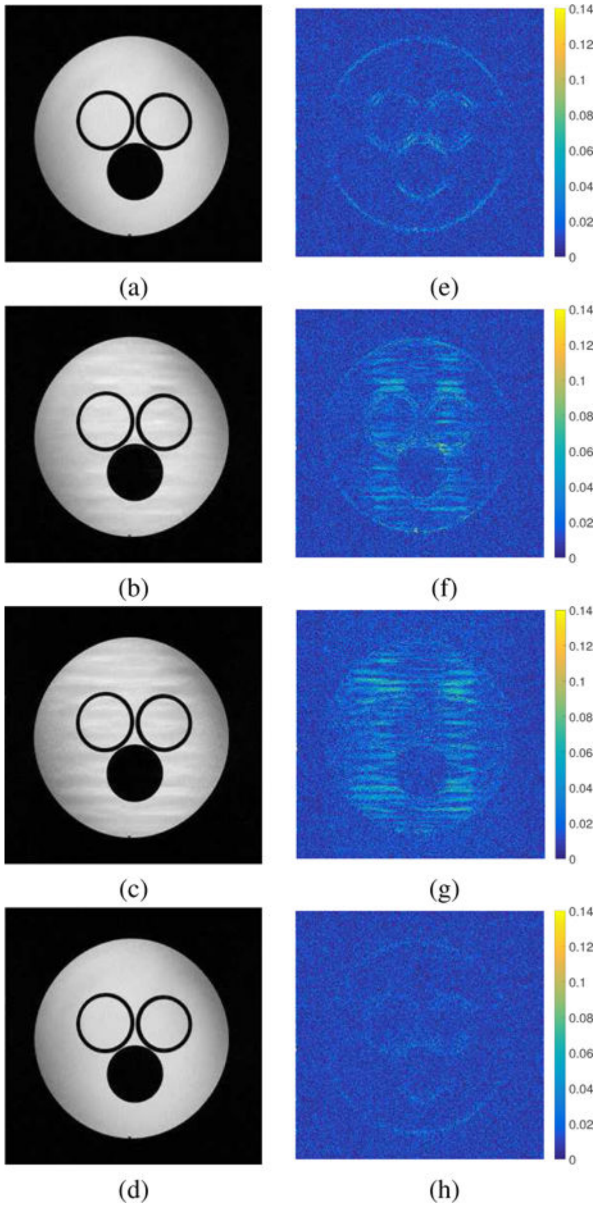


Fig. 6.

Results for Image (c) with Cartesian sampling and 2.5× undersampling. The sampling mask is shown in Fig. 5(a). Reconstructions (magnitudes): (a) DLMRI [13]; (b) PANO [75]; (c) ℓ_0 “norm”-based FDLCP [18]; and (d) SOUP-DILLO MRI (with zero-filling initialization). (e)-(h) are the reconstruction error maps for (a)-(d), respectively.

TABLE I

PSNRS corresponding to the zero-filling (initiaL $\mathbf{y}^0 = \mathbf{A}^{\dagger}\mathbf{z}$), Sparse MRI [6], PANO [75], DLMRI [13], SOUP-DILLI MRI (for (P4)), and SOUP-DILLO MRI (for (P3)) reconstructions for various images. The simulated undersampling factors (UF), and k-space undersampling schemes are listed for each example. The best PSNRs are marked in bold. The image labels are as per Fig. 3.

Image	Sampling	UF	Zero-filling	Sparse MRI	PANO	DLMRI	SOUP-DILLI MRI	SOUP-DILLO MRI
a	Cartesian	7x	27.9	28.6	**31.1**	**31.1**	30.8	**31.1**
b	Cartesian	2.5x	27.7	31.6	41.3	40.2	38.5	**42.3**
c	Cartesian	2.5x	24.9	29.9	34.8	36.7	36.6	**37.3**
c	Cartesian	4x	25.9	28.8	**32.3**	32.1	32.2	**32.3**
d	Cartesian	2.5x	29.5	32.1	36.9	38.1	36.7	**38.4**
e	Cartesian	2.5x	28.1	31.7	40.0	38.0	37.9	**41.5**
f	2D random	5x	26.3	27.4	30.4	30.5	30.3	**30.6**
g	Cartesian	2.5x	32.8	39.1	41.6	41.7	42.2	**43.2**

TABLE II

PSNRS corresponding to the ℓ_0 "NORM"-based FDLCP reconstructions [18], and the SOUP-DILLO MRI (for (P3)) reconstructions obtained with a zero-filling ($\mathbf{y}^0 = \mathbf{A}^\dagger \mathbf{z}$) initialization or by initializing with the FDLCP result (last column). The various images, sampling schemes, and undersampling factors (UF) are the same as in Table I. The best PSNRs are marked in bold.

Image	UF	FDLCP (ℓ_0 "norm")	SOUP-DILLO MRI (Zero-filling init.)	SOUP-DILLO MRI (FDLCP init.)
a	7x	**31.5**	31.1	**31.5**
b	2.5x	44.2	42.3	**44.8**
c	2.5x	33.5	**37.3**	**37.3**
c	4x	32.8	32.3	**33.5**
d	2.5x	38.5	38.4	**38.7**
e	2.5x	43.4	41.5	**43.9**
f	5x	30.4	**30.6**	**30.6**
g	2.5x	43.2	43.2	**43.5**