A GREEDY FORWARD-BACKWARD ALGORITHM FOR ATOMIC NORM CONSTRAINED MINIMIZATION

Nikhil Rao *Parikshit Shah* *Stephen Wright* *Robert Nowak*

University of Wisconsin - Madison

ABSTRACT

In many applications in signal and image processing, communications, and system identification, one aims to recover a signal that has a simple representation in a given basis or frame. Key devices for obtaining such representations are objects called atoms, and functions called atomic norms. These concepts unify the idea of simple representations across several known applications, and motivate extensions to new problem classes of interest. In important special cases, fast and efficient algorithms are available to solve the reconstruction problems, but an approach that works well for the general atomic-norm paradigm has not been forthcoming to date. In this paper, we combine a greedy selection scheme with a backward step that sparsifies the basis by removing less significant elements that were included at earlier iterations. We show that the overall scheme achieves the same convergence rate as the forward greedy scheme alone, provided that backward steps are taken only when they do not degrade the solution quality too badly. Finally, we validate our method by describing applications to three problems of interest.

Index Terms— Algorithms, Atomic Norm, Greedy Approximation, Compressed Sensing

1. INTRODUCTION

Minimization of a convex loss function with a constraint on the "simplicity" of the solution has found widespread applications in communications, learning, image processing, genetics, and other fields. While obvious formulations of the simplicity requirement are intractable, there are sometimes tractable, convex formulations available. The notion of simplicity varies across applications: For many applications in signal and image processing, we wish the recovered high-dimensional signal to be *sparse*. In matrix-completion problems that arise in recommendation systems, we seek *low-rank* solution matrices. Convex relaxations of the sparsity and low-rank constraints [1, 2, 3] lead to tractable ℓ_1- and nuclear-norm-constrained optimization programs, respectively. In image processing and multitask learning applications, the optimal vector/image is known to be group sparse in a certain representation, leading to formulations as group-lasso-norm constrained problems (with or without overlap between the groups) [4, 5, 6]. In applications involving finite rates of innovation [7], signals are known to lie in a union of subspaces. The group lasso norm can be modified to yield a penalty that constrains the target variables to lie in such unions [8].

Since these formulations differ so markedly across applications, Chandrasekaran et al. [9] study the question of whether there is a principled, unified way to derive the best convex heuristic for different notions of simplicity. The notions of *atoms* and *atomic norms* provides such a framework. We define atomic norms and several of their applications in Section 2.

While atomic norms lead to good heuristics for *formulating* reconstruction problems, efficient algorithms to *solve* these optimization formulations remains a challenge. For special cases such as ℓ_1-constrained [10, 11, 12] and nuclear-norm-constrained [6, 13, 14] formulations, highly efficient special-purpose algorithms are known. To extend such frameworks to general atomic norms, the authors of [15] introduced a greedy method based on the Frank-Wolfe algorithm [16]. The algorithm employs a *forward greedy scheme*, in which a single atom is added to the basis at each iteration. This approach suffers the drawback that errors made in previous iterations cannot be quickly erased in subsequent iterations, and that atoms, once added to the basis, cannot be removed or replaced by more suitable choices. The alternative approach of *backward greedy selection* [17], which starts with the full atomic set and removes an atom from the model at each iteration, is often impractical, since the full set of atoms is very large or uncountably infinite.

In this paper, we propose a *Forward-Backward* scheme that adds atoms greedily in the same manner as in [15], while allowing atoms to be purged later if their contribution to the observations is superseded by atoms selected later in the process. Our scheme also admits flexibility in adjusting the current iterate, for example, by sparsifying its representation in terms of the current basis. Our algorithm enjoys similar convergence properties to the method of [15] (as we can show by making minor adjustments to the analysis of that paper) while producing solutions that are significantly sparser, as can be seen from our experiments.

We apply our method to a standard compressed sensing formulation and to two other problems of current interest: moment problems and latent group lasso. In moment problems [18, 19], which arise in applications such as radar, communications, seismology, and sensor arrays, one aims to recover frequency components of a received sampled signal. The possible frequencies lie on a continuum, making the atomic set uncountably infinite. As shown in [18], the corresponding atomic norm problem can be reformulated as a semidefinite program (SDP) in certain cases. However, general-purpose SDP solvers are expensive for this problem, and impractical for large instances. Our forward-backward method, coupled with a "repeated random discretization" strategy, recovers unknown frequency components from the signal with high accuracy and reasonable computational efficiency. Our approach has the additional advantage that it does not rely on rationality of the sampling times.

The latent group lasso [20] arises from applications in genomics, image processing, and machine learning [4, 6]. It is shown in [20, 8] that the latent group lasso penalty, which is a sum of ℓ_2 norms of overlapping groups of variables, can be modeled as an atomic norm. However, solving the problem involves replication of the variables and solution of a higher dimensional problem [6], which can become expensive when the amount of overlap between groups is significant. When applied to this problem, our forward-backward algorithm does not require variable replication and can be implemented efficiently.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

2. NOTATIONS AND PROBLEM SETUP

We use boldface letters $\boldsymbol{x}, \boldsymbol{y}$ etc. to denote variables in the problem. For example, in the case of sparse signal recovery (or group sparse), $\boldsymbol{x} \in \mathbb{R}^p$ represents a vector. In matrix completion applications, $\boldsymbol{x} \in \mathbb{R}^{m \times n}$ represents a matrix.

We assume the existence of a known atomic set $\mathcal{A}$, containing elements that lie in the same space as the problem variables. The atoms $\boldsymbol{a} \in \mathcal{A}$ form the basic building blocks of signals of interest. A variable $\boldsymbol{x}$ may be representable as a conic combination of atoms $\boldsymbol{a} \in \mathcal{A}_t$ in a subset $\mathcal{A}_t \subset \mathcal{A}$, as follows:

$$\boldsymbol{x} = \sum_{\boldsymbol{a} \in \mathcal{A}_t} c_{\boldsymbol{a}} \boldsymbol{a}, \quad \text{with } c_{\boldsymbol{a}} \geq 0 \text{ for all } \boldsymbol{a} \in \mathcal{A}_t, \tag{1}$$

where the $c_{\boldsymbol{a}}$ are scalar coefficients. We write $\boldsymbol{x} \in \text{co}(\mathcal{A}_t, \tau)$ for some given $\tau \geq 0$, if it is possible to represent the vector $\boldsymbol{x}$ in the form (1), with the additional constraint

$$\sum_{\boldsymbol{a} \in \mathcal{A}_t} c_{\boldsymbol{a}} \leq \tau. \tag{2}$$

Given an $\boldsymbol{x} \in \mathbb{R}^p$ and an atomic set, we define the *atomic norm*:

$$\|\boldsymbol{x}\|_{\mathcal{A}} = \inf \left\{ \sum_{\boldsymbol{a} \in \mathcal{A}} c_a : \boldsymbol{x} = \sum_{\boldsymbol{a} \in \mathcal{A}} c_a \boldsymbol{a}, \;\; c_a \geq 0 \;\; \forall \boldsymbol{a} \in \mathcal{A} \right\}, \tag{3}$$

that is, $\|\boldsymbol{x}\|_{\mathcal{A}}$ is derived from the representation of x in terms of the full set $\mathcal{A}$ of atoms with the smallest coefficient sum. Note that the coefficient sum (2) is an upper bound on the atomic norm $\|x\|_{\mathcal{A}}$; different coefficients or a different subset of atoms may allow $\boldsymbol{x}$ to be expressed using a smaller sum of coefficients.

In a typical linear inverse problem, we look to recover a simple representation of $\boldsymbol{x}$ (as a conic combination of a modest number of atoms) from linear measurements of the form $\boldsymbol{y} = \boldsymbol{\Phi x}$. A provably effective way to obtain a simple $\boldsymbol{x}$ is to place a constraint on its atomic norm; see [9]. We formulate this problem using an ℓ_2 loss function as follows:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi x}\|_2^2 \;\; \text{subject to} \;\; \|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau. \tag{4}$$

The rest of the paper is organized as follows. In Section 3, we describe our forward-backward algorithm to solve (4), and analyze its convergence properties in Section 4, proving a $1/T$ convergence rate. In Section 5, we describe various experiments on real and simulated data. We conclude and discuss future research in Section 6.

3. ALGORITHM

The forward step of our algorithm below chooses a new atom and adjusts the coefficients to the basis using the same strategy as in [15]. The backward step removes one of the existing basis elements if the value of f is not degraded by more than a certain fraction (less than 1) of the improvement gained during the forward step. All iterates remain feasible with respect to the constraint $\|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau$.

We give more detail on the crucial steps. Step 9 allows the vector $\hat{\boldsymbol{x}}_{t+1}$ obtained from the atom selection and coefficient updating procedure of [15] to be replaced by another vector $\tilde{\boldsymbol{x}}_{t+1}$ that is expressible in terms of the same basis $\mathcal{A}_{t+1}$, satisfies atomic-norm constraint, and has a lower function value. This step is optional; it suffices for the analysis to simply set $\tilde{\boldsymbol{x}}_{t+1}$ to the value $\hat{\boldsymbol{x}}_{t+1}$ obtained in Step 8. Alternatively, we can take some steps of a descent algorithm, such as projected gradient [21], starting from this point.

Algorithm 1 Forward Backward algorithm for Atomic Norm Minimization

```
 1: Input: Characterization of $\mathcal{A}$, Bound $\tau$, Backward Parameter
    $0 < \eta < 1$;
 2: Initialize Choose some $\boldsymbol{a}_0 \in \mathcal{A}$, set $\boldsymbol{x}_0 = \tau \boldsymbol{a}_0$, set $\mathcal{A}_0 = \{\boldsymbol{a}_0\}$,
    set $t \leftarrow 0$;
 3: repeat
 4:   FORWARD STEP
 5:   $\boldsymbol{a}' \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{a} \rangle$;
 6:   $\mathcal{A}_{t+1} \leftarrow \mathcal{A}_t \cup \boldsymbol{a}'$;
 7:   $\gamma_{t+1} \leftarrow \arg\min_{\gamma \in [0,1]} f(\boldsymbol{x}_t + \gamma(\tau \boldsymbol{a}' - \boldsymbol{x}_t))$;
 8:   $\hat{\boldsymbol{x}}_{t+1} \leftarrow \boldsymbol{x}_t + \gamma_{t+1}(\tau \boldsymbol{a}' - \boldsymbol{x}_t)$;
 9:   Find any $\tilde{\boldsymbol{x}}_{t+1} \in \text{co}(\mathcal{A}_{t+1}, \tau)$ with $f(\tilde{\boldsymbol{x}}_{t+1}) \leq f(\hat{\boldsymbol{x}}_{t+1})$;
10:   Express $\tilde{\boldsymbol{x}}_{t+1} = \sum_{\boldsymbol{a} \in \mathcal{A}_{t+1}} c_{\boldsymbol{a}} \boldsymbol{a}$;
11:   BACKWARD STEP
12:   Find the term $c_{\boldsymbol{a}''} \boldsymbol{a}''$ such that $f(\tilde{\boldsymbol{x}}_{t+1} - c_{\boldsymbol{a}} \boldsymbol{a})$ is minimized
      over all $\boldsymbol{a} \in \mathcal{A}_{t+1}$;
13:   Find any $\bar{\boldsymbol{x}}_{t+1} \in \text{co}(\mathcal{A}_{t+1} \setminus \boldsymbol{a}'', \tau)$ such that $f(\bar{\boldsymbol{x}}_{t+1}) \leq$
      $f(\boldsymbol{x}_{t+1} - c_{\boldsymbol{a}''} \boldsymbol{a}'')$;
14:   if $[f(\bar{\boldsymbol{x}}_{t+1}) - f(\tilde{\boldsymbol{x}}_{t+1})] \leq \eta[f(\boldsymbol{x}_t) - f(\tilde{\boldsymbol{x}}_{t+1})]$ then
15:     $\mathcal{A}_{t+1} \leftarrow \mathcal{A}_{t+1} \setminus \boldsymbol{a}''$;
16:     $\boldsymbol{x}_{t+1} \leftarrow \bar{\boldsymbol{x}}_{t+1}$;
17:     Express $\boldsymbol{x}_{t+1} = \sum_{\boldsymbol{a} \in \mathcal{A}_{t+1}} c_{\boldsymbol{a}} \boldsymbol{a}$;
18:   else
19:     $\boldsymbol{x}_{t+1} \leftarrow \tilde{\boldsymbol{x}}_{t+1}$;
20:   end if
21:   $t \leftarrow t + 1$;
22: until convergence
```

Similarly, Step 13 allows the sparsified iterate $\boldsymbol{x}_{t+1} - c_{\boldsymbol{a}''}\boldsymbol{a}''$ to be replaced by any point $\bar{\boldsymbol{x}}_{t+1}$ with the same basis set that satisfies the atomic-norm constraint and has a lower function value. Again, this step is optional; we can simply set $\bar{\boldsymbol{x}}_{t+1} \leftarrow \boldsymbol{x}_{t+1} - c_{\boldsymbol{a}''}\boldsymbol{a}''$. Alternatively, we can perform steps of gradient projection to the problem of minimizing $f(\sum_{\boldsymbol{a} \in \mathcal{A}_{t+1} \setminus \boldsymbol{a}''} c_{\boldsymbol{a}} \boldsymbol{a})$ over the simplex defined by $\sum_{\boldsymbol{a} \in \in \mathcal{A}_{t+1} \setminus \boldsymbol{a}''} c_{\boldsymbol{a}} \leq \tau$, $c_{\boldsymbol{a}} \geq 0$, using the technique in [21].

Notice that the selection of the sparsifying atom in the backward step — Step 12 — can be performed efficiently. From the form of f defined in (4), we have

$$f(\boldsymbol{x}_{t+1} - c_{\boldsymbol{a}}\boldsymbol{a}) = f(\boldsymbol{x}_{t+1}) - c_{\boldsymbol{a}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{a} \rangle + \frac{1}{2} c_{\boldsymbol{a}}^2 \|\Phi \boldsymbol{a}\|_2^2, \tag{5}$$

and the quantities $\|\Phi \boldsymbol{a}\|_2^2$ can be computed efficiently and stored as soon as each atom $\boldsymbol{a}$ enters the current basis $\mathcal{A}_t$.

Steps 10 and 17 call for the coefficients $c_{\boldsymbol{a}}$ to be consistent with the current value of $\boldsymbol{x}$ and the current atomic basis set.

By varying the parameter η we can control the frequency of backward steps. A value closer to 1 will yield more frequent removal of atoms. In Section 5, we fix $\eta = 1/2$, in the spirit of [22].

It is possible for the atom $\boldsymbol{a}'$ added in the forward step of some iteration t to be immediately removed in the backward step of the same iteration. This can happen because the reoptimizations in Steps 10 and 13 may identify a different set of coefficients for the same basis $\mathcal{A}_t$ that improves the objective, and the new atom is not required. This behavior could indicate that the existing basis is adequate to describe an approximate solution, that no new atoms are needed, and that a near-optimal solution has been found. Our implementation terminates when this behavior happens repeatedly.

4. ANALYSIS

The analysis of our algorithm is a straightforward modification of [15], so we provide only a sketch that highlights the points of difference. Two definitions are needed.

Definition [15, Definition 1]. Given a function $f(\cdot)$, a norm $\|\cdot\|$ and a set S, we define

$$L_{\|\cdot\|}(f,S) := \sup_{\boldsymbol{x},\boldsymbol{y}\in S,\ \ \alpha\in(0,1]} \frac{f((1-\alpha)\boldsymbol{x}+\alpha\boldsymbol{y}) - f(\boldsymbol{x}) - \langle\nabla f(\boldsymbol{x}),\alpha(\boldsymbol{y}-\boldsymbol{x})\rangle}{\alpha^2\|\boldsymbol{y}-\boldsymbol{x}\|^2}$$

Given $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}\|^2$, we have $L_{\|\cdot\|}(f,S) \le \|\boldsymbol{\Phi}^T\boldsymbol{\Phi}\|$.

Definition [15]. $R := \sup_{\boldsymbol{a}\in\mathcal{A}}\|\boldsymbol{a}\|$.

Theorem 4.1. *Assume that $f(\cdot)$ is convex and smooth. Let $\boldsymbol{x}^\star$ be the optimum value attained by Algorithm 1. Suppose we initialize the algorithm with $\boldsymbol{x}_0$, and let $\eta := \frac{1}{2}$, $L := L_{\|\cdot\|}(f,S)$, and R be defined as above. Then the iterates from Algorithm* 1 *converge according to the following: after T iterations, we have*

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \le \frac{2(B+2L\tau^2R^2)^2}{BT},$$

where $B = f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)$.

We use $\eta = 1/2$ only for simplicity; a similar convergence rate can be obtained for any $\eta\in(0,1)$ by a simple modification of the proof.

Proof. Recall that $\hat{\boldsymbol{x}}_{t+1} = \boldsymbol{x}_t + \gamma_{t+1}(\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t)$ and $\tilde{\boldsymbol{x}}_{t+1}$ (both generated in the forward step) are feasible with respect to the atomic-norm constraint and satisfy $f(\tilde{\boldsymbol{x}}_{t+1}) \le f(\hat{\boldsymbol{x}}_{t+1})$. We thus have

$$\begin{aligned} f(\tilde{\boldsymbol{x}}_{t+1}) - f(\boldsymbol{x}_t) &\le f(\hat{\boldsymbol{x}}_{t+1}) - f(\boldsymbol{x}_t) \\ &\le \min_{\gamma\in[0,1]}\left(2\gamma^2L\tau^2R^2 - \gamma(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star))\right), \end{aligned}$$

where the second inequality follows from the analysis in [15]. Noting that $\eta = 1/2$, we take a backward step only if

$$f(\bar{\boldsymbol{x}}_{t+1}) - f(\tilde{\boldsymbol{x}}_{t+1}) \le \frac{1}{2}\left(f(\boldsymbol{x}_t) - f(\tilde{\boldsymbol{x}}_{t+1})\right). \qquad (6)$$

By combining these bounds, we have

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) &\le \frac{1}{2}(f(\tilde{\boldsymbol{x}}_{t+1}) - f(\boldsymbol{x}_t)) \\ &\le \frac{1}{2}\min_{\gamma\in[0,1]}\left(2\gamma^2L\tau^2R^2 - \gamma(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star))\right). \end{aligned}$$

By a recursive application of this bound, we obtain

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \le (2(B+2L\tau^2R^2)^2)/(BT) \qquad (7)$$

□

5. EXPERIMENTS AND RESULTS

In all our experiments, we simply set $\tilde{\boldsymbol{x}}_{t+1} \leftarrow \hat{\boldsymbol{x}}_{t+1}$ in Step 9. For the backward step, we choose the atom $\boldsymbol{a}''$ to delete (Step 12) according to (5), and perform projected gradient iterations to update the coefficients $c_{\boldsymbol{a}}$, for $\boldsymbol{a}\in\mathcal{A}_{t+1}\setminus\boldsymbol{a}''$ (Step 13). In each of our experiments, we set τ clairvoyantly, since we have access to the true signal. In practice, τ can be chosen using standard methods such as cross validation.

5.1. Sparse Signal Recovery

We tested our method on the compressed sensing framework:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|^2 \quad \text{s.t. } \|\boldsymbol{x}\|_1 \le \tau$$

We consider a sparse signal of length $p = 500$, with $k = 20$ nonzeros. We obtain 80 $(4\times k)$ i.i.d. Gaussian measurements, and corrupt the measurements with AWGN of standard deviation $\sigma = 0.1$. We set $\tau = 1.1 \times \|\boldsymbol{x}_{true}\|_1$, where $\boldsymbol{x}_{true}$ is the true signal. We fixed the maximum number of iterations to be 200.

Fig 1 compares our method (F-B) to that of [15]. We also compare our method to CoSaMP [23], another method that employs a backward or pruning step. We see that our method recovers a sparser signal, and has a better MSE value for fit to the true signal. Note that after the optimization step 13, we set some coefficients to 0. In Table 1, we compare the runtimes of our method to that of Frank-Wolfe. We see that although we take backward steps, our method takes significantly less time as the problem size increases.

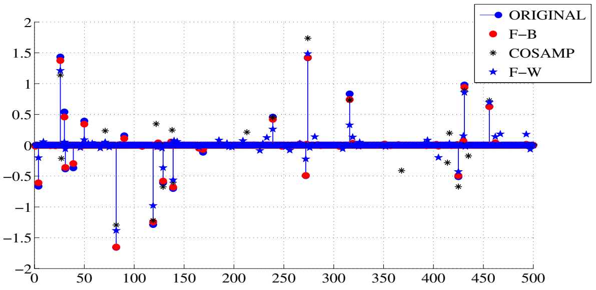


Fig. 1. Comparison of algorithms for ℓ_1-regularized least squares. The F-B method (MSE $\approx 1.22\times10^{-4}$) outperforms Frank-Wolfe (F-W) [15] (MSE $\approx$ 0.035), and CoSaMP (MSE $\approx$ 0.0050). In 200 iterations, we take 124 backward steps to remove elements from the basis, leaving 76 elements in the final basis. Of these, 72 were unique and only 25 have nonzero coefficients. The corresponding values for F-W [15] are 75 and 60, respectively

# variables	**F-B**	**F-W**
512	2.1	38.8
1024	3.2	122
2048	23.3	1298

Table 1. Runtimes (in seconds) of Forward-Backward (F-B) and the Frank-Wolfe (F-W) methods. For a problem with p variables, we take $p/4$ Gaussian measurements of a $p/16$ sparse vector. Convergence tolerance is 10^{-6} and maximum iterations is $2p$.

5.2. Moment Problems in Signal Processing

Consider a continuous time signal

$$\phi(t) = \sum_{j=1}^{k} c_j\exp(i2\pi f_j t),$$

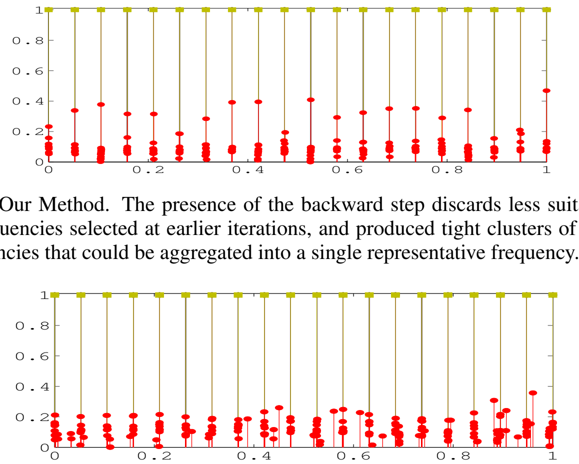
where each $f_j\in[0,1]$. In many applications of interest, $\phi(t)$ is sampled at times $S := \{t_1, t_2, \ldots, t_n\}$ giving an observation vector

$\boldsymbol{x} := [\phi(t_1), \phi(t_2), \ldots, \phi(t_n)] \in \mathbb{C}^n$. We thus have

$$\boldsymbol{x} = \sum_{j=1}^{k} c_j a(f_j) \text{ where } a(f_j) = \left[e^{i2\pi f_j t_1} \ \ldots \ e^{i2\pi f_j t_n} \right]^T.$$

In [18], the authors consider the case where $S \subset \{1, \ldots, N\}$, a random integer-valued subset over some sampling horizon N. The Fourier transform of $\phi(t)$ can be viewed as a signed measure supported on $[0, 1]$ and the acquired sample vector x is a (partial) trigonometric moment vector with respect to this measure. Reconstructing the measure — finding the unknown coefficients c_j and frequencies f_j from $\boldsymbol{x}$ — is a challenging problem in general. A natural convex relaxation analyzed in [18] to the problem is (4) with $\Phi = I$, where the atomic norm is with respect to the atoms $a(f)$ described above.

In applying our algorithm to this problem, a minor technical hurdle is Step 5, which involves finding the maximum modulus of a trigonometric polynomial on the unit circle. While solvable as a semidefinite program [24], we propose a simpler "random-gridding" approach, in which several freqencies are chosen at random, and the one with the most suitable frequency among these is selected as the new atom. Due to the ability of our method to purge irrelevant atoms, we can replace atoms picked in earlier iterations by more suitable atoms identified at later iterations.



(a) Our Method. The presence of the backward step discards less suitable frequencies selected at earlier iterations, and produced tight clusters of frequencies that could be aggregated into a single representative frequency.

(b) The method of [15], with random gridding. Many false positive frequencies are selected.

Fig. 2. Compressed Sensing off the Grid. The yellow squares are the true frequencies, and the red circles are those estimated by greedy methods.

We generated a signal consisting of a sum of 20 frequency components, spaced equally between $(0, 1]$. We obtained $n = 60$ samples of the signal. We ran both our algorithm and that of [15] for 200 iterations. At each iteration, we obtain 1000 points, with frequencies chosen randomly from the interval $(0, 1]$. Fig. 2(a) shows the performance of our method. Note that, the repeated random gridding allows us to recover the frequency components accurately. Also, note that the backward step is key to the success of this method. By contrast, we see in Fig. 2(b) that many spurious frequencies remain in the reported solution if the algorithm contains no backward step.

An interesting possible modification of our algorithm would be to optimize the trigonometric polynomial via *adaptive gridding*, rather than the simple randomized gridding that is performed here

5.3. Group Sparse Models in Wavelet based Signal Processing

The authors of [20] propose the latent group lasso, a method that recovers signals whose support can be expressed as a union of groups. That the penalty can be expressed as an atomic norm is shown in [20, 8]. In [8], the authors use the concept to group parent-child pairs in DWT coefficients, and perform image recovery. We see that our method serves as the "greedy" analogue of the latent group lasso. The method of this paper does not require replication of variables (as was done in [4]), and hence avoids the inflation of problem dimension associated with replication.

We consider some standard 1D signals [25], and aim to recover the parent child DWT coefficients modeled into groups. In each case, we considered a length 1024 signal, and obtained 300 Gaussian measurements corrupted with AWGN $\sigma = 0.01$. Each signal was scaled to lie between ±1, and we restricted ourselves to 200 iterations of the algorithm. Fig 3 shows that we recover a piecewise polynomial signal fairly accurately. MSE results for other test signals are shown in Table 2. We set $\tau = 1.1 \times \sum_G \|\boldsymbol{x}_G\|$, where $\boldsymbol{x}_G$ is the Haar DWT of the true signal restricted to the indices in group G.

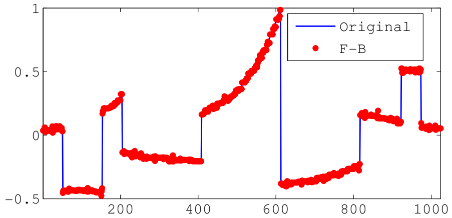


Fig. 3. Recovery of the Piecewise Polynomial test signal using Parent-Child DWT coefficient groupings.

Signal	MSE F-B	MSE Forward Greedy
Piece Polynomial	$\mathbf{1.38 \times 10^{-4}}$	2.767×10^{-4}
Blocks	$\mathbf{2.126 \times 10^{-4}}$	7.593×10^{-4}
HeaviSine	**0.0021**	0.0023
Piecewise Regular	**0.0028**	0.0083

Table 2. Recovery of some 1d test signals in the presence of AWGN ($\sigma = 0.01$). We see that at the end of 200 iterations, our method consistently outperforms forward greedy selection [15]

6. CONCLUSIONS AND FURTHER RESEARCH

We have presented a forward-backward scheme for atomic-norm constrained minimization. We showed that our method works better than the simple forward greedy selection. The backward step makes use of the quadratic form of the objective function to decide efficiently on which atom to remove from the current basis.

In future work, we will investigate reoptimization methods in the forward and backward steps, that is, effective and efficient implementations of steps 9 and 13 that may exploit the properties of the underlying applications. For example, we will use some steps of gradient projection over the simplex, and incorporate adaptive gridding strategies in the application to moment problems. We may also consider performing more than one backward step on each iteration.

7. REFERENCES

[1] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Information Theory*, vol. 52, pp. 489–509, 2006.

[2] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, pp. 717–772, 2009.

[3] B. Recht, "A simple approach to matrix completion," *Journal of Machine Learning Research*, vol. 12, pp. 3413–3430, 2011.

[4] N. Rao, R. Nowak, S. Wright, and N. Kingsbury, "Convex approaches to model wavelet sparsity patterns," *IEEE Conference on Image Processing*, pp. 1917–1920, 2011.

[5] F. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, June 2008.

[6] L. Jacob, G. Obozinski, and J. P. Vert, "Group lasso with overlap and graph lasso," *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[7] M. V. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE transactions on Signal Processing*, vol. 50, pp. 1417–1428, Jun 2002.

[8] N. Rao, B. Recht, and R. Nowak, "Signal recovery in unions of subspaces with applications to compressive imaging," *Preprint arXiv:1209.3079*, 2012.

[9] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. Willsky, "The convex geometry of linear inverse problems," *preprint arXiv:1012.0621v1*, 2010.

[10] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2479–2493, 2009.

[11] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 49–67, 2006.

[12] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society. Series B*, pp. 267–288, 1996.

[13] R. Jenatton, J. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.

[14] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society. Series B*, vol. 68, pp. 49–67, 2006.

[15] A. Tewari, P. Ravikumar, and I. Dhillon, "Greedy algorithms for structurally constrained high dimensional problems," *Advances in Neural Information Processing Systems*, vol. 24, pp. 882–890, 2011.

[16] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 2006.

[17] C. Couvreur and Y. Bresler, "On the optimality of the backward greedy algorithm for the subset selection problem," *SIAM Journal of Matrix Analysis and Applications*, vol. 21, pp. 797–808, 2000.

[18] G. Tang, B. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," *Preprint arXiv:1207.6053*, 2012.

[19] E. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Preprint arXiv:1203.5871*, 2012.

[20] G. Obozinski, L. Jacob, and J. Vert, "Group LASSO with overlaps: The latent group LASSO approach," *Preprint arXiv:1110.0413v1 [stat.ML]*, Oct 2011.

[21] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 272–279.

[22] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Transactions on Information Theory*, vol. 57, pp. 4689–4708, 2011.

[23] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[24] B. Dumitrescu, *Positive trigonometric polynomials and signal processing applications*. Springer, 2007.

[25] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.