Forward - Backward Greedy Algorithms for Atomic Norm Regularization

Nikhil Rao† Parikshit Shah# Stephen Wright⋆
†Department of Electrical and Computer Engineering
Wisconsin Institute for Discovery
⋆ Department of Computer Sciences
University of Wisconsin - Madison

***Abstract*—In many signal processing applications, one aims to reconstruct a signal that has a simple representation with respect to a certain basis or frame. Fundamental elements of the basis known as "atoms" allow us to define "atomic norms" that can be used to construct convex regularizers for the reconstruction problem. Efficient algorithms are available to solve the reconstruction problems in certain special cases, but an approach that works well for general atomic norms remains to be found. This paper describes an optimization algorithm called CoGEnT , which produces solutions with succinct atomic representations for reconstruction problems, generally formulated with atomic-norm constraints. CoGEnT combines a greedy selection scheme based on the conditional gradient approach with a backward (or "truncation") step that exploits the quadratic nature of the objective to reduce the basis size. We establish convergence properties and validate the algorithm via extensive numerical experiments on a suite of signal processing applications. Our algorithm and analysis are also novel in that they allow for *inexact* forward steps. In practice, CoGEnT significantly outperforms the basic conditional gradient method, and indeed many methods that are tailored to specific applications, when the truncation steps are defined appropriately. We also introduce several novel applications that are enabled by the atomic-norm framework, including tensor completion, moment problems in signal processing, and graph deconvolution.**

I. Introduction

Minimization of a convex loss function with a constraint on the "simplicity" of the solution has found widespread applications in communications, machine learning, image processing, genetics, and other fields. While exact formulations of the simplicity requirement are often intractable, it is sometimes possible to devise tractable formulations via convex relaxation that are (nearly) equivalent. Since these formulations differ so markedly across applications, a principled and unified convex heuristic for different notions of simplicity has been proposed using notions of *atoms* and *atomic norms* [1]. Atoms are fundamental basis elements of the representation of a signal, chosen so that "simplicity" equates to "representable in terms of a small number of atoms." We list several applications, describing for each application a choice of atoms that captures the concept of simplicity for those applications.

For instance, a sparse signal x may be represented as $x = \sum_{j \in \mathcal{S}} c_j e_j$, where the e_j are the standard unit vectors and $\mathcal{S}$ captures the support of x. One can view the set $\{\pm e_j\}$ as *atoms* that constitute the signal, and the convex hull of these atoms is a set of fundamental importance called the *atomic-norm ball*. The operation of inflation/deflation of the atomic norm ball induces a norm (the *atomic norm*), which serves as an effective regularizer (see Sec. I-A for details). The atomic set $\{\pm e_j,\ j = 1, 2, \ldots, p\}$ induces the ℓ_1 norm [2], now well-known to be an effective regularizer for sparsity. However, this idea can be generalized. For instance, the atomic norm induced by the convex hull of all unit rank matrices is the nuclear norm, often used as a heuristic for rank minimization [3], [4]. Other novel applications of the atomic-norm framework include the following.

- **Group-norm-constrained multitask learning** problems with group-ℓ_2 norms [5]–[7] or group-ℓ_∞ norms [8]–[10] have as atoms unit Euclidean balls and unit ℓ_∞-norm balls, respectively, restricted to specific groups of variables.
- **Group lasso with overlapping groups** arises from applications in genomics, image processing, and machine learning [5], [7]. It is shown in [11] that the sum of ℓ_2 norms of overlapping groups of variables is an atomic norm.
- **Moment problems**, which arise in applications such as radar, communications, seismology, and sensor arrays, have an atomic set which is uncountably infinite [12]. Each atom is a trigonometric

moment sequence of an atomic measure supported on the unit interval [12]. This methodology can be extended to signal classes such as Bessel functions, Gaussians, and wavelets.

- **Group testing on graphs** and network tomography finds widespread applications in sensor, computer, social, and biological networks [7], [13]. In such applications, it is typically required to identify a set of faulty edges/nodes from measurements that are based on the known structure of the graph. Each atom can be defined as a subset of nodes or edges in the graph.
- **Hierarchical norms** arise in topic modeling [14], climate and oceanology applications [15], and fMRI data analysis [16]. The atoms here are hybrids of group-sparse and sparse atoms.
- **OSCAR**-regularized problems use an octagonal penalty to simultaneously identify a sparse set of pairwise correlated variables [17]. The atoms are vectors containing at most two nonzeros, with each nonzero entry being ± 1.
- **Tensor Completion:** Signals modeled as tensors have recently enjoyed renewed interest in machine learning [18]. In the case we consider here, in which the tensor is symmetric, orthogonally decomposable, and low (symmetric) rank, the atoms consist of unit-rank symmetric tensors.
- **Deconvolution** is the problem of splitting a signal $z = x + y$ into its constituent components x and y [19], where x and y are succinct with respect to different sets of atoms. Typical cases include the atomic sets being sparse and low rank [20], sparse in the canonical and discrete cosine transform (DCT) bases, and sparse and group sparse [21].

We present a general method called CoGEnT (for "Conditional Gradient with Enhancement and Truncation") that can be applied to general atomic norm problems, in particular to all the applications discussed above. CoGEnT reconstructs signals by minimizing a least-squares loss function that measures the difference between the signal representation and the observations, subject to a "simplicity" constraint on the signal, imposed in terms of an atomic norm. Besides its generality, novel aspects of CoGEnT include (a) introduction of *enhancement steps* at each iteration to improve solution fidelity, (b) introduction of efficient *backward steps* that dramatically improves the performance, (c) introduction of the notion of *inexactness* in the forward step. A comprehensive convergence result is presented.

A. Preliminaries and Notation

We assume the existence of a known atomic set $\mathcal{A}$ and an unknown signal $\boldsymbol{x}$ in some "ambient" space, where $\boldsymbol{x}$ is a superposition of a small number of atoms from $\mathcal{A}$. (We emphasize that the set of atoms need not be finite.) We assume further that the set $\mathcal{A}$ is symmetric about the origin, that is, $\boldsymbol{a} \in \mathcal{A} \Rightarrow -\boldsymbol{a} \in \mathcal{A}$. The representation of $\boldsymbol{x}$ as a conic combination of atoms $\boldsymbol{a} \in \mathcal{A}_t$ in a subset $\mathcal{A}_t \subset \mathcal{A}$ is written 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

$$\boldsymbol{x} \in \text{co}(\mathcal{A}_t, \tau), \tag{2}$$

for some given $\tau \geq 0$, if it is possible to represent the vector $\boldsymbol{x}$ in the form (1), with the additional constraint

$$\sum_{\boldsymbol{a} \in \mathcal{A}_t} c_{\boldsymbol{a}} \leq \tau. \tag{3}$$

We use $\boldsymbol{A}_t$ to denote a linear operator which maps the coefficient vector c (with cardinality $|\mathcal{A}_t|$) to a vector in the ambient space, using the vectors in $\mathcal{A}_t$, that is,

$$\boldsymbol{A}_t c := \sum_{\boldsymbol{a} \in \mathcal{A}_t} c_{\boldsymbol{a}} \boldsymbol{a}. \tag{4}$$

Since there is a one-to-one relationship between $\mathcal{A}_t$ and the linear operator $\boldsymbol{A}_t$, we use the notation (4) more often, and sometimes slightly abuse terminology by referring to $\boldsymbol{A}_t$ as the "basis" at iteration t. We sometimes refer to the "columns" of $\boldsymbol{A}_t$, by which we mean the elements of the corresponding basis $\mathcal{A}_t$.

The *atomic norm* [1] is the gauge functional induced by $\mathcal{A}$:

$$\|\boldsymbol{x}\|_{\mathcal{A}} := \inf\{t > 0 : \boldsymbol{x} \in t(\text{conv}(\mathcal{A}))\}, \tag{5}$$

where conv $(\cdot)$ denotes the convex hull of a collection of points. Equivalently, we have

$$\|\boldsymbol{x}\|_{\mathcal{A}} := \inf \left\{ \sum_{\boldsymbol{a} \in \mathcal{A}} c_a : \boldsymbol{x} = \sum_{\boldsymbol{a} \in \mathcal{A}} c_a \boldsymbol{a}, \quad c_a \geq 0 \right\}. \tag{6}$$

Given a representation (1), the sum of coefficients in (3) is an *upper bound* on the atomic norm $\|x\|_{\mathcal{A}}$. The dual atomic norm is given by

$$\|\boldsymbol{x}\|_{\mathcal{A}}^* = \sup_{\|\boldsymbol{u}\|_{\mathcal{A}} \leq 1} \langle \boldsymbol{u}, \boldsymbol{x} \rangle. \tag{7}$$

The dual atomic norm is key to our approach — the atom selection step in CoGEnT amounts to choosing the argument that achieves the supremum in (7), for a particular choice of $\boldsymbol{x}$.

Our algorithm CoGEnT solves the convex optimization problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau, \qquad (8)$$

where $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w}$ corresponds to observed measurements, with noise vector $\boldsymbol{w}$. The regularizing constraint on the atomic norm of $\boldsymbol{x}$ enforces "simplicity" with respect to the chosen atomic set. Efficient algorithms are known for this problem when the atoms are standard unit vectors $\pm e_j$ (for which the atomic norm is the ℓ_1 norm) [22]–[24] and rank-one matrices (for which the atomic norm is the nuclear norm) [7], [25], [26]. CoGEnT targets the general formulation (8), opening up a suite of new applications with rigorous convergence guarantees and state-of-the-art empirical performance.

We remark that while (8) is a convex formulation, tractable algorithms for solving it are not known in full generality. Indeed, characterization of the atomic norm is itself intractable in some cases. From an optimization perspective, interior point methods are often impractical, being either difficult to formulate or too slow for large-scale instances. First order greedy methods are often the methods of choice. Greedy schemes are popular in high dimensional signal recovery settings because of their computational efficiency, scalability to large datasets, and interesting global rate-of-convergence properties. They have found widespread use in large scale machine learning applications [27]–[33].

B. Past Work: Conditional Gradient Method

A conditional gradient (CG) algorithm for (8) was introduced in [29]. This greedy approach is often known as "Frank-Wolfe" after the authors who proposed it in the 1950s [34]. At each iteration, it finds the atom that optimizes a first-order approximation to the objective over the feasible region, and adds this atom to the basis for the solution. Each iteration of CoGEnT performs a "forward step" of this type. Although CoGEnT includes various enhancements, it is this forward step that drives the convergence theory, which is similar to that of standard conditional gradient methods [28], [29], although with a different treatment of inexactness in the choice of search direction.

Although greedy methods require more iterations than such prox-linear methods as SpaRSA [22], FISTA [35], and Nesterov's accelerated gradient method [36], each iteration is typically less expensive. For example, in matrix completion applications, prox-linear methods require computation of an SVD of a matrix [37] (or at least a substantial part of it), while CG requires only the computation of the leading singular vectors. In other applications, such as structural SVM [38], CG schemes are the only practical way to solve the optimization formulation. Latent group lasso [7] can be extended to perform regression on very large signals by employing a "replication" strategy, but as the amount of group overlap increases, prox-linear methods quickly become memory intensive. CG offers a scalable method to solve problems of this form. The procedure to choose each new atom has a linear objective, as opposed to the quadratic program required to perform projection steps in prox-linear methods. The linear subproblem is often easier to solve; in some applications, it makes the difference between tractability and intractability. Moreover, the linear problem need only be solved approximately to retain convergence guarantees [27], [31].

C. Backward (Truncation) Steps

In signal processing applications, one is interested not only in minimizing the loss function, but also in the "simplicity" of the solutions. For example, when the solution corresponds to the wavelet coefficients of an image, sparsity of the representation is key to its usefulness as a compact representation. In this regard, the basic CG and indeed all greedy schemes suffer from a significant drawback: atoms added at some iterations may be superseded by others added at later iterations, and ultimately may not contribute much to reducing the loss function. By the time the loss function has been reduced to an acceptable level, the basis may contain many such atoms of dubious usefulness, thus detracting from the quality of the solution.

Backward steps in CoGEnT allow atoms to be removed from the basis when they are found to be unhelpful in reducing the objective. We define this step in a flexible way, the only requirement being that it does not degrade the objective function too greatly in comparison to the gain that was obtained at the most recent "forward" iteration. The enhancement / reoptimization step discussed below is one way to perform truncation; we can simply discard those atoms whose coefficients are reduced to zero when we reoptimize over the current basis. This step may be expensive to implement, so we seek alternatives. One such alternative is to test one-by-one the effect of removing each atom in the current basis — an operation that can be performed efficiently because of the least-squares nature of the loss function in (8) — and remove the atom(s) that do not deteriorate the objective beyond a specified limit. A third alternative is to seek a completely new set of basis atoms that can be combined to obtain a vector with similar objective value to the latest iteration.

We note that the backward steps in CoGEnT are quite different from the "away steps" analyzed in [27], [39]. These steps move in the opposite to the "worst possible" linearized direction, and thus *add* a new element to the basis at each iteration, rather than *removing* elements, as we do here. While away steps have been shown to improve the convergence properties of CG method, they do not contribute to enhancing sparsity of the solution.

Forward-backward greedy schemes for ℓ_1 constrained minimization have been considered previously in [40]–[43]. These methods build on the Orthogonal Matching Pursuit (OMP) algorithm [23], and cannot be readily extended to the general setting (8).

D. Enhancement (Reoptimization) Steps

The enhancement / reoptimization step in CoGEnT takes the current basis and seeks a new set of coefficients in the representation (4) that reduces the objective while satisfying the norm constraint. (A "full correction" step of this type was described in [27].) The step is implemented as a linear least-squares objective over a simplex. CoGEnT solves it with a gradient projection method, using a warm start based on the current set of coefficients. Projection onto the simplex can be performed in $O(n_{t+1})$ operations, where n_{t+1} is the dimension of the simplex (which equals the number of elements in the current basis $\mathcal{A}_{t+1}$). Since gradient projection is a descent method that maintains feasibility, it can be stopped after any number of iterations, without prejudice to the convergence of CoGEnT.

E. Outline of the Paper

The rest of the paper is organized as follows. We specify CoGEnT in the next section, describing different variants of the backward step that promote parsimonious solutions (involving small numbers of atoms). In Section III, we state convergence results, deferring proofs to an appendix. Section IV describes the application of CoGEnT to a number of existing applications, and compares it to various other methods that have been proposed for these applications. In Section V, we apply CoGEnT for a variety of *new* applications, for which current methods, if they exist at all, do not scale well to large data sets. In Section VI we extend our algorithm to deal with deconvolution problems.

II. Algorithm

CoGEnT is specified in Algorithm 1. Its three major elements — the forward (conditional gradient) step, the backward (truncation) step, and the enhancement (reoptimization) step — have been discussed in Section I. We note that these three steps are constructed so that the iterates at each step are feasible (that is, $\|\boldsymbol{x}_t\|_{\mathcal{A}} \leq \tau$). We make further notes in this section about alternative implementations of these three steps.

Algorithm 1 CoGEnT: Conditional Gradient with Enhancement and Truncation

1: **Input:** Characterization of $\mathcal{A}$, bound τ, acceptance threshold $\eta \in (0, 1/2]$;
2: **Initialize**, $\boldsymbol{a}_0 \in \mathcal{A}$, $t \leftarrow 0$, $\boldsymbol{A}_0 \leftarrow [\boldsymbol{a}_0]$, $c_0 \leftarrow [\tau]$, $\boldsymbol{x}_0 \leftarrow \boldsymbol{A}_0 c_0$;
3: **repeat**
4: $\quad \boldsymbol{a}_{t+1} \leftarrow \arg\min_{\boldsymbol{a}\in\mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{a}\rangle$; {FORWARD}
5: $\quad \tilde{\boldsymbol{A}}_{t+1} \leftarrow [\boldsymbol{A}_t \;\; \boldsymbol{a}_{t+1}]$;
6: $\quad \gamma_{t+1} \leftarrow \arg\min_{\gamma\in[0,1]} f(\boldsymbol{x}_t + \gamma(\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t))$; {LINE SEARCH}
7: $\quad \tilde{c}_{t+1} \leftarrow [(1-\gamma_{t+1})c_t \;\; \gamma_{t+1}\tau\boldsymbol{a}_{t+1}]$;
8: $\quad$ **Optional:** Approximately solve $\tilde{c}_{t+1} \leftarrow \arg\min_{c_{t+1}} f(\tilde{\boldsymbol{A}}_{t+1}c_{t+1})$ s.t. $\|c_{t+1}\|_1 \leq \tau$, $c_{t+1} \geq 0$ with the output from Step 7 as a warm start; {ENHANCEMENT}
9: $\quad \tilde{\boldsymbol{x}}_{t+1} = \tilde{\boldsymbol{A}}_{t+1}\tilde{c}_{t+1}$;
10: $\quad$ Threshold $F_{t+1} := \eta f(\boldsymbol{x}_t) + (1-\eta) f(\tilde{\boldsymbol{x}}_{t+1})$;
11: $\quad [\boldsymbol{A}_{t+1}, c_{t+1}, x_{t+1}]$ $= \text{TRUNCATE}(\tilde{A}_{t+1}, \tilde{c}_{t+1}, \tau, F_{t+1})$; {BACKWARD}
12: $\quad t \leftarrow t+1$;
13: **until convergence**
14: **Output:** $\boldsymbol{x}_t$

The forward step (Step 4) is equivalent to solving an approximation to (8) based on a linearization of f around the current iterate. Specifically, it is easy to show that $\tau\boldsymbol{a}_t$ solves the following problem:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}_t) + \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x} - \boldsymbol{x}_t\rangle \quad \text{s.t.} \quad \|\boldsymbol{x}\|_{\mathcal{A}} \leq \tau.$$

(A simple argument reveals that the minimizer of this problem is attained by $\tau\boldsymbol{a}$, where $\boldsymbol{a}$ is an atom.) For many applications of interest, this step can be performed efficiently, often more efficiently than the corresponding projection/shrinkage step in prox-linear methods.

The line search of Step 6 can be performed exactly, because of the quadratic objective in (8). We obtain

$$\gamma_{t+1} = \min\left\{ \frac{\langle \boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}_t, \boldsymbol{\Phi}\boldsymbol{v}\rangle}{\|\boldsymbol{\Phi}\boldsymbol{v}\|^2}, 1 \right\}, \; \boldsymbol{v} := \tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t.$$

We now discuss two options for performing the backward (truncation) step (Step 11), whose purpose is to compactify the representation of $\boldsymbol{x}_t$, without degrading

the objective more than a specified amount. The parameter η defines a sufficient decrease criterion that the modified solution needs to satisfy. A value of η closer to its upper bound will yield more frequent removal of atoms and hence a sparser solution, at the expense of more modest progress per iteration.

Our first implementation of the truncation step seeks to purge one or more elements from the expanded basis $\boldsymbol{A}_{t+1}$, using a quadratic prediction of the effect of removal of each atom in turn. The approach is outlined in Algorithm 2. Removal of an atom $\boldsymbol{a}$ from the current iterate $\tilde{\boldsymbol{x}}_{t+1}$ in Step 4 of Algorithm 2 would result in the following change to the objective:

$$\begin{aligned} & f(\tilde{\boldsymbol{x}}_{t+1} - c_{\boldsymbol{a}}\boldsymbol{a}) \\ & \quad = f(\tilde{\boldsymbol{x}}_{t+1}) - c_{\boldsymbol{a}}\langle \nabla f(\tilde{\boldsymbol{x}}_{t+1}), \boldsymbol{a}\rangle + \frac{1}{2}c_{\boldsymbol{a}}^2 \|\boldsymbol{\Phi}\boldsymbol{a}\|_2^2. \end{aligned} \tag{9}$$

(We have assumed that $c_{\boldsymbol{a}}$ is the coefficient of $\boldsymbol{a}$ in the current representation of $\tilde{\boldsymbol{x}}_{t+1}$.) The quantities $\|\boldsymbol{\Phi}\boldsymbol{a}\|_2^2$ can be computed efficiently and stored as soon as each atom $\boldsymbol{a}$ enters the current basis $\boldsymbol{A}_t$, so the main cost in evaluating this criterion is in forming the inner product $\langle \nabla f(\tilde{\boldsymbol{x}}_{t+1}), \boldsymbol{a}\rangle$. Having chosen a candidate atom that optimizes the degradation in f, we can reoptimize over the remaining elements (Step 6 in Algorithm 2), possibly using the same gradient-projection approach as in Step 8 of Algorithm 1), and test to see whether the updated value of f still falls below the threshold F_{t+1}. Note that Step 6 in Algorithm 2 is optional; we could alternately define by $\hat{c}_{t+1}$ by removing the coefficient corresponding to the discarded atom from c_{t+1}. Atom removal is repeated in Algorithm 2 as long as the successively updated objective stays below the threshold F_{t+1}.

Our second implementation of the truncation step allows for a wholesale redefinition of the current basis, seeking a new, smaller basis and a new set of coefficients such that the objecive value is not degraded too much. The approach is specified in Algorithm 3. It is motivated by the observation that atoms added at early iterates contain spurious components, which may not be cancelled out by atoms added at later iterations. This phenomenon is apparent in matrix completion, where the number of atoms (rank-one matrices) generated by the procedure above is often considerably larger than the rank of the target matrix. For this application, we could implement Algorithm 3 by forming a singular value decomposition of the matrix represented by the latest iterate $\tilde{\boldsymbol{x}}_{t+1}$, and defining a new basis $\hat{\boldsymbol{A}}_{t+1}$ to be the rank-one matrices that correspond to the largest singular values. These singular values would then form the new coefficient vector $\hat{c}_{t+1}$, and the new iterate $\boldsymbol{x}_{t+1}$ would be defined in terms of just these singular values and singular vectors. The computational work required for such a step would be comparable with one iteration of the popular singular value thresholding (SVT) approach [37] for matrix completion, which also requires calculation of the leading singular values and singular vectors.

Algorithm 2 : TRUNCATE($\tilde{A}_{t+1}, \tilde{c}_{t+1}, \tau, F_{t+1}$)

1: **Input:** Current basis $\tilde{\boldsymbol{A}}_{t+1}$, coefficient vector $\tilde{c}_{t+1}$, iterate $\tilde{\boldsymbol{x}}_{t+1} = \tilde{A}_{t+1}\tilde{c}_{t+1}$; bound τ; threshold F_{t+1};
2: continue $\leftarrow 1$;
3: **while** continue$= 1$ **do**
4: $\quad \hat{\boldsymbol{a}}_{t+1} \leftarrow \arg\min_{\boldsymbol{a}\in\tilde{\boldsymbol{A}}_{t+1}} f(\tilde{\boldsymbol{x}}_{t+1} - c_{\boldsymbol{a}}\boldsymbol{a})$
5: $\quad \hat{\boldsymbol{A}}_{t+1} \leftarrow \tilde{\boldsymbol{A}}_{t+1}\backslash\{\hat{\boldsymbol{a}}_{t+1}\}$;
6: $\quad$ Find $\hat{c}_{t+1} \geq 0$ with $\|\hat{c}_{t+1}\|_1 \leq \tau$ such that $f(\hat{\boldsymbol{A}}_{t+1}\hat{c}_{t+1}) \leq f(\tilde{\boldsymbol{x}}_{t+1} - (\tilde{c}_{\tilde{\boldsymbol{a}}_{t+1}})_{t+1}\hat{\boldsymbol{a}}_{t+1})$;
7: $\quad$ **if** $f(\hat{A}_{t+1}\hat{c}_{t+1}) \leq F_{t+1}$ **then**
8: $\quad\quad \tilde{\boldsymbol{A}}_{t+1} \leftarrow \hat{\boldsymbol{A}}_{t+1}$;
9: $\quad\quad \tilde{\boldsymbol{x}}_{t+1} \leftarrow \hat{\boldsymbol{A}}_{t+1}\hat{c}_{t+1}$;
10: $\quad\quad \tilde{c}_{t+1} \leftarrow \hat{c}_{t+1}$;
11: $\quad$ **else**
12: $\quad\quad$ continue $\leftarrow 0$;
13: $\quad$ **end if**
14: **end while**
15: $\boldsymbol{A}_{t+1} \leftarrow \tilde{\boldsymbol{A}}_{t+1}$; $\boldsymbol{x}_{t+1} \leftarrow \tilde{\boldsymbol{x}}_{t+1}$; $c_{t+1} \leftarrow \tilde{c}_{t+1}$;
16: **Output:** Possibly reduced basis $\boldsymbol{A}_{t+1}$, coefficient vector $c_{t+1} \geq 0$, and iterate $\boldsymbol{x}_{t+1}$.

Algorithm 3 TRUNCATE($\tilde{A}_{t+1}, \tilde{c}_{t+1}, \tau, F_{t+1}$)

1: **Input:** Current basis $\tilde{\boldsymbol{A}}_{t+1}$, coefficient vector $\tilde{c}_{t+1}$, iterate $\tilde{\boldsymbol{x}}_{t+1} = \tilde{A}_{t+1}\tilde{c}_{t+1}$; bound τ; threshold F_{t+1};
2: Find alternative basis $\hat{\boldsymbol{A}}_{t+1}$ and coefficients $\hat{c}_{t+1} \geq 0$ such that $\#columns(\hat{\boldsymbol{A}}_{t+1}) < \#columns(\tilde{\boldsymbol{A}}_{t+1})$, $\|\hat{c}_{t+1}\|_1 \leq \tau$;
3: **if** $f(\hat{\boldsymbol{A}}_{t+1}\hat{c}_{t+1}) \leq F_{t+1}$ **then**
4: $\quad \boldsymbol{A}_{t+1} \leftarrow \hat{\boldsymbol{A}}_{t+1}$; $\boldsymbol{x}_{t+1} \leftarrow \hat{\boldsymbol{A}}_{t+1}\hat{c}_{t+1}$; $c_{t+1} \leftarrow \hat{c}_{t+1}$;
5: **else**
6: $\quad \boldsymbol{A}_{t+1} \leftarrow \tilde{\boldsymbol{A}}_{t+1}$; $\boldsymbol{x}_{t+1} \leftarrow \tilde{\boldsymbol{x}}_{t+1}$; $c_{t+1} \leftarrow \tilde{c}_{t+1}$;
7: **end if**
8: **Output:** Possibly reduced basis $\boldsymbol{A}_{t+1}$, coefficient vector $c_{t+1} \geq 0$, and iterate $\boldsymbol{x}_{t+1}$.

We conclude this section by discussing practical stopping criteria for Algorithm 1. As we show in Section III, CoGEnT is guaranteed to converge to an optimum, and the objective is guaranteed to decrease at each iteration.

We therefore use the following termination criteria:

$$f(\boldsymbol{x}_{t+1}) \leq \text{tol}, \quad \text{or} \quad \frac{f(\boldsymbol{x}_{t-1}) - f(\boldsymbol{x}_t)}{f(\boldsymbol{x}_{t-1})} \leq \text{tol},$$

where tol is a small user-defined parameter.

III. Convergence Results

Convergence properties for CoGEnT are stated here, with proofs appearing in the appendix. Sublinear convergence of CoGEnT (Theorem III.1) follows from a mostly familiar argument.

Theorem III.1. *Consider the convex optimization problem* (8)*, and let $\boldsymbol{x}^*$ be a solution of* (8)*. Let $\eta \in (0, 1/2]$. Then the sequence of function values $\{f(\boldsymbol{x}_t)\}$ generated by CoGEnT converges to $f^* = f(\boldsymbol{x}^*)$ with*

$$f(\boldsymbol{x}_T) - f^* \leq \frac{\bar{C}}{T+1}, \quad \textit{for all } T \geq 1, \qquad (10)$$

where

$$\begin{aligned}
\bar{C}_1 &:= \eta D + 2(1-\eta)LR^2\tau^2, \\
\bar{C} &:= \frac{2\bar{C}_1^2}{(1-\eta)(\bar{C}_1 - LR^2\tau^2)} > 0, \\
L &:= \|\boldsymbol{\Phi}^T\boldsymbol{\Phi}\|, \\
D &:= f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*), \\
R &:= \max_{\boldsymbol{a}\in\mathcal{A}} \|\boldsymbol{a}\|.
\end{aligned}$$

When the true optimum $\boldsymbol{x}^\star$ lies in the interior of the set $\|\boldsymbol{x}\| \leq \tau$, and when $\boldsymbol{\Phi}$ has full row rank, we can obtain linear convergence using ideas that are similar in spirit to those used in [44] for the standard CG method. We omit the formal statement and full proof of this result, since in most applications of interest, the solution will lie on the boundary of the atomic-norm ball.

Similar convergence properties hold when the atom added in the forward step of Algorithm 1 is computed *approximately*[1]. In place of the $\arg\min$ in Step 4 of Algorithm 1, we have the following requirement on $\boldsymbol{a}_{t+1} \in \mathcal{A}$:

$$\langle \nabla f(\boldsymbol{x}_t), (\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t)\rangle \leq (1-\omega) \min_{\boldsymbol{a}\in\mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \tau\boldsymbol{a} - \boldsymbol{x}_t\rangle \qquad (11)$$

where $\omega \in (0, 1/4)$ is a user-defined parameter. Note that (11) implies that $\langle \nabla f(\boldsymbol{x}_t), \tau\boldsymbol{a}_{t+1}\rangle \leq (1-\omega)\min_{\boldsymbol{a}\in\mathcal{A}}\langle \nabla f(\boldsymbol{x}_t), \tau\boldsymbol{a}\rangle + \omega\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t\rangle$ so that this condition essentially requires us to find a solution of the Frank-Wolfe subproblem with relative objective accuracy ω. If a lower bound for the minimum is available from duality, this condition can be checked in practice. This criterion is similar in spirit to the inexact Newton method for nonlinear equations [45, pp. 277-279], which requires the approximate solution of the linearized model to achieve only a fraction of the decrease promised by exact solution of the model.

For the relaxed definition (11) of $\boldsymbol{a}_{t+1}$, we obtain the following result.

Theorem III.2. *Assume that the conditions of Theorem III.1 hold, but that the atom $\boldsymbol{a}_{t+1}$ selected in Step 4 in Algorithm 1 satisfies the condition* (11)*. Assume further than $\eta \in (0, 1/3)$ and $\omega \in (0, 1/4)$. Then we have*

$$f(\boldsymbol{x}_T) - f^* \leq \frac{\tilde{C}}{T+1} \quad \textit{for all } T \geq 1, \qquad (12)$$

where

$$\begin{aligned}
\tilde{C}_1 &:= (\eta + \omega(1-\eta))D + 2(1-\eta)LR^2\tau^2, \\
\tilde{C} &:= \frac{2\tilde{C}_1^2}{(1-\eta)[(1-\omega)\tilde{C}_1 - LR^2\tau^2]},
\end{aligned}$$

with L, R, τ, D defined as in Theorem III.1

IV. Experiments: Standard Applications in Sparse Recovery

CoGEnT can be used to solve a variety of problems from signal processing and machine learning. We describe some experiences with such problems.

A. Sparse Signal Recovery

We tested our method on the following compressed sensing formulation:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^p} \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \quad \text{s.t. } \|\boldsymbol{x}\|_1 \leq \tau. \qquad (13)$$

The atoms in this case are the signed canonical basis vectors, and the atom selection (Step 4 in Algorithm 1) reduces to the following:

$$\begin{aligned}
\hat{i} &= \arg\max_i |[\nabla f(\boldsymbol{x}_t)]_i|, \\
\boldsymbol{a}_{t+1} &= -\text{sign}\left([\nabla f(\boldsymbol{x}_t)]_{\hat{i}}\right) e_{\hat{i}}.
\end{aligned}$$

We consider a sparse signal $\boldsymbol{x}$ of length $p = 20000$, with 5% of coefficients randomly assigned values from $\mathcal{N}(0, 1)$. Setting $n = 5000$, we construct the $n \times p$ matrix $\boldsymbol{\Phi}$ to have i.i.d. Gaussian entries, and corrupt the measurements with Gaussian noise (AWGN) of standard deviation $\sigma = 0.01$. In the formulation (13), we set $\tau = \|\boldsymbol{x}^\star\|_1$, where $\boldsymbol{x}^\star$ is the chosen optimal signal.

To check the performance of CoGEnT against the conditional gradient method, we run both methods for a

[1]Approximately solving this step can give substantial gains in practicality of the algorithm, making the method useful in a wider variety of applications, as we see in later sections

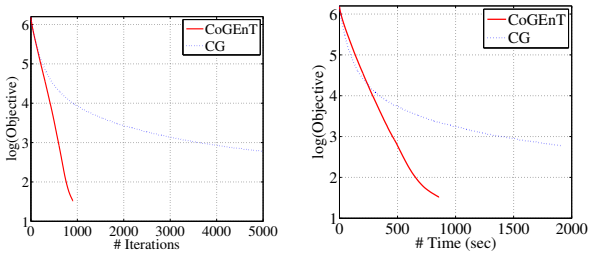


Fig. 1: Comparison between CoGEnT and standard conditional gradient (CG).

maximum of 5000 iterations, with a stopping tolerance of 10^{-8}. Figure 1 shows a graph of the logarithm of the function value vs iteration count (left) and logarithm of the function value vs wall clock time (right). On a per-iteration basis, CoGEnT performs more operations than standard CG. However, the backward steps yield faster reduction in the objective function value, resulting in better convergence, even when measured in terms of run time.

Figure 2 shows a comparison of solution quality obtained by CoGEnT, CG, CoSaMP [32], and Subspace Pursuit [33]. As a performance metric, we used both the mean square error and the Hamming Distance between the true and predicted vectors. We performed 10 independent trials, setting $\boldsymbol{\Phi}$ in each trial to be a 1000×5000 matrix, with reference solution $\boldsymbol{x}^\star$ chosen to have $s = 200$ nonzeros. Observations $\boldsymbol{y}$ were corrupted with AWGN with standard deviation σ in the range $[0, 2]$. In CoGEnT and CG, we chose $\tau := \|\boldsymbol{x}^\star\|_1$. For CoSaMP and the Subspace Pursuit methods, we set $s = 200$, the known sparsity level of the optimal signal $\boldsymbol{x}^\star$. Figure 2 shows the results.

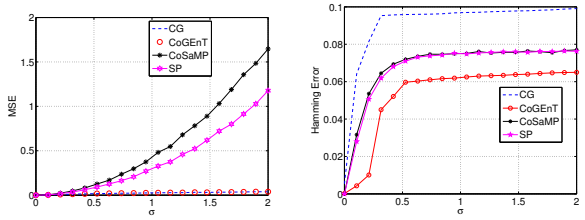


Fig. 2: Comparison of solution quality obtained by different methods. Left: MSE for recovered solution as a function of observation noise parameter σ. Right: Hamming error in recovered solution as a function of σ.

B. Overlapping Group Lasso

In group-sparse variants of (13) we seek vectors $\boldsymbol{x}$ such that $\boldsymbol{\Phi x} \approx \boldsymbol{y}$ for given $\boldsymbol{\Phi}$ and $\boldsymbol{y}$, such that the support of $\boldsymbol{x}$ consists of a small number of predefined groups of the coefficients. We denote each group by $G \subset \{1, 2, \ldots, p\}$ and denote the full collection of groups by $\mathcal{G}$. CG and CoGEnT do not require replication of variables, as is done in prox-linear algorithms [5], [7]. The atom selection step (Step 4 in Algorithm 1) amounts to the following operation:

$$\hat{G} = \arg\max_{G \in \mathcal{G}} \|[\nabla(f(\boldsymbol{x}_t))]_G\|,$$
$$[\boldsymbol{a}_{t+1}]_{\hat{G}} = -[\nabla f(\boldsymbol{x}_t))]_{\hat{G}} / \|[\nabla f(\boldsymbol{x}_t))]_{\hat{G}}\|$$
$$[\boldsymbol{a}_{t+1}]_i = 0 \ \text{ for } i \notin \hat{G}.$$

We compare the performance of CoGEnT with an accelerated prox-linear (PL) approach [22] that uses variable replication. We considered M group sparse signals with $\lfloor M/10 \rfloor$ groups chosen to be active in the reference solution, where each group has size 50. The groups are ordered in linear fashion with the last 30 indices of each group overlapping with the first 30 of the next group. We then took $n = \lceil p/2 \rceil$ measurements with a Gaussian sensing matrix $\boldsymbol{\Phi}$, with AWGN of standard deviation $\sigma = 0.1$ added to the observations. Table I shows runtimes for the two approaches.

M	True Dimension	Replicated Dimension	time CoGEnT	time PL
100	2030	5000	15.	22.
1000	20030	50000	211.	462.
1200	24030	60000	359.	778.
1500	30030	75000	575.	1377.
2000	40030	100000	852.	2977.

TABLE I: Recovery times (in seconds) for CoGEnT and prox-linear methods applied to a synthetic overlapping group-sparse problem.

C. Matrix Completion

In low-rank matrix completion, the atoms are rank-one matrices and the observations are individual elements of the matrix. If $\boldsymbol{u}$, $\boldsymbol{v}$ are the first left and right singular vectors of $-\nabla f_t$, the solution of Step 4 in Algorithm 1 is $\boldsymbol{a}_{t+1} = \boldsymbol{u}\boldsymbol{v}^T$. The cost of finding only the top singular vectors in the gradient matrix is usually much lower than performing the full SVD.

V. Experiments: Novel Applications

We now report on the application of CoGEnT to recovery problems in several novel areas of application. In some cases, CoGEnT and CG are the only practical approaches for solving these problems.

A. Tensor Completion

Recovery of low-rank tensor approximations arises in applications ranging from multidimensional signal processing to latent-factor models in machine learning [18]. We consider the recovery of symmetric orthogonal tensors from incomplete measurements using CoGEnT. We seek a tensor T of the form $T = \sum_{i=1}^{r} c_i[\otimes \boldsymbol{u}_i]$, where $\otimes \boldsymbol{u}$ indicates an t-fold tensor product of a vector $\boldsymbol{u} \in \mathbb{R}^p$. We obtain partial measurements of this tensor of the form $y = \mathcal{M}(T)$, where $\mathcal{M}(\cdot)$ is a *masking operator* that reveals a certain subset of the entries of the tensor. We formulate this problem in an atomic norm setup, wherein the objective function that captures the data fidelity term is $f(T) := \frac{1}{2}\|y - \mathcal{M}(T)\|^2$. The atomic set has the form

$$\mathcal{A} = \{\otimes \boldsymbol{u} : \boldsymbol{u} \in \mathbb{R}^p,\ \|\boldsymbol{u}\|_2 = 1\}.$$

In applying CoGEnT to this problem, the greedy step requires calculation of the symmetric rank-one tensor that best approximates the gradient of the loss function. This calculation can be performed efficiently using power iterations [18]. We implement a backward step based on basis reoptimization and thresholding (Algorithm 3), where the new basis is obtained from a tensor decomposition, computed via power iterations.

We look to recover toy $10 \times 10 \times 10$ tensors, with 50% of the entries observed using CoGEnT (without noise). Figure 4 shows accuracy of recovery for tensors of various ranks. While the recovered tensor does not always match the rank of the original tensor, it does indeed have low rank and small component-wise error. We declare that recovery is "exact" if each entry of the recovered tensor is within 10^{-3} relative error w.r.t. the original tensor. We used a (relative) stopping tolerance of 10^{-6}, running the method for a maximum of 100 iterations.

Fig. 3 shows the phase transition plot for performing tensor recovery for random $20 \times 20 \times 20$ tensors, using different fractions of sampled entries.

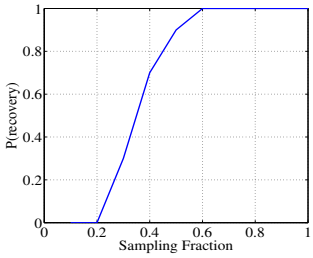


Fig. 3: Recovery vs fraction of observations.

Rank	MSE
2	5.406×10^{-5}
3	3.4789×10^{-4}
4	4.999×10^{-5}
5	5.4929×10^{-4}

Fig. 4: Accuracy of tensors recovered, from 50% of exact observations.

B. Moment Problems in Signal Processing

Consider a continuous time signal

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

for frequencies $f_j \in [0, 1]$, $j = 1, 2, \ldots, k$ and coefficients $c_j > 0$, $j = 1, 2, \ldots, k$. In many applications of interest, $\phi(t)$ is sampled at times $S := \{t_i\}_{i=1}^{n}$ giving an observation vector $\boldsymbol{x} := [\phi(t_1), \phi(t_2), \ldots, \phi(t_n)] \in \mathbb{C}^n$. The observed information is therefore

$$\boldsymbol{x} = \sum_{j=1}^{k} c_j a(f_j),$$

where

$$a(f_j) = \left[e^{i2\pi f_j t_1}, e^{i2\pi f_j t_2}, \ \ldots, e^{i2\pi f_j t_n}\right]^T.$$

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 [12], is obtained by setting $\boldsymbol{\Phi} = I$ in (8) and defining the atoms to be $a(f)$ for $f \in [0, 1]$, a set of infinite cardinality.

The main technical issue in applying CoGEnT to this problem is the greedy atom selecion step (Step 4 of Algorithm 1), which requires us to find the maximum modulus of a trigonometric polynomial on the unit circle. This operation can be formulated as a semidefinite program [46], but since SDPs do not scale well to high dimensions [12], this approach has limited appeal. In our implementation of CoGEnT, we form a discrete grid of frequency values. We start with an initial grid of equally spaced frequencies, then refine it between iterations by adding new frequencies midway between each pair of selected frequencies. By controlling the discretization in this way, we are essentially controlling the inexactness of the forward step. Indeed, the accuracy requires in (11) can provide guidance for the adaptive discretization process. Step 4 simply selects an atom $a(f)$ corresponding to the frequency f in the current grid that forms the most negative inner product with the gradient of the loss function.

Our implementation of the backward step for this problem has two parts. Besides performing Algorithm 2 to remove multiple uninteresting frequencies, we include a heuristic for merging nearby frequencies, replacing multiple adjacent spikes by a single spike, when it does not degrade the fit to observations too much to do so.

Fig. 5 compares the performance of CoGEnT with that of standard CG on a signal with ten randomly chosen frequencies in $[0, 1]$. We take samples at 300 timepoints of a signal of length 1000, corrupted with AWGN with

standard deviation .01. The left figure in Figure 5 shows the signal recovered by CoGEnT, indicating that all but the smallest of the ten spikes were recovered accurately. The critical role played by the backward step can be seen by contrasting these results with those reported for CG in the right figure of Fig. 5, where many spurious frequencies appear.

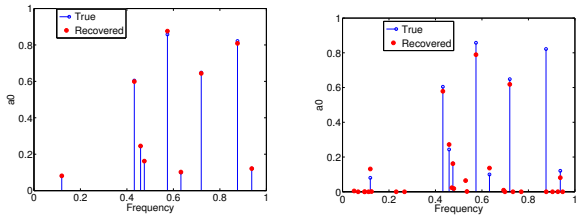


Fig. 5: CoGEnT and CG for off-grid compressed sensing. Blue spikes and circles represent the reference solution, and red circles are those estimated by the algorithms.

We compared CoGEnT to the SDP formulation as explained in [12]. Although the SDP solves the problem exactly, it does not scale well to large dimensions, as we show in the timing comparisons of Figure 6.

The formulation above can be generalized to include signals that are a conic combination of a few arbitrary functions of the form $\phi(t, \alpha_i)$.

- Bessel and Airy functions form natural signal ensembles that arise as solutions to differential equations in physics. As an example, letting $J_r(t)$ denoting Bessel functions of the first kind, we have

$$\phi(t; \alpha_1, \alpha_2, \alpha_3) = J_{\alpha_1}\left(\frac{t}{\alpha_2} - \alpha_3\right),$$

  where $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}_+$. Here, each atom is defined by a specific choice of the triple $(\alpha_1, \alpha_2, \alpha_3)$. (Again, the atomic set $\mathcal{A}$ has infinite cardinality.)
- Triangle and sawtooth waves. Consider for instance

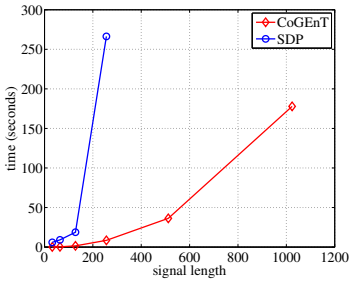
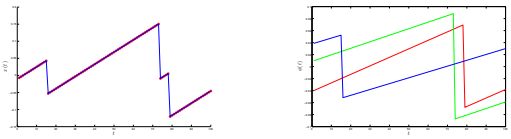


Fig. 6: Speed comparison with SDP. The SDP formulation does not scale well.



(a) The true signal (blue) is a superposition of sawtooth functions. Red dots show samples acquired.

(b) Sawtooth components recovered by CoGEnT.

Fig. 7: Recovering sawtooth components by sampling. (Best seen in color)

the sawtooth functions:

$$\phi(t; \alpha_1, \alpha_2) = \frac{t}{\alpha_1} - \left\lfloor \frac{t}{\alpha_1} \right\rfloor - \alpha_2,$$

where $\alpha_1, \alpha_2 \in \mathbb{R}_+$. Each atom is defined by a specific choice of (α_1, α_2). Figure 7 shows successful recovery of a superposition of sawtooth functions from a limited number of samples.

- Ricker wavelets arise in seismology applications, with the atoms characterized by $\sigma > 0$:

$$\phi(t; \sigma) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}}\left(1 - \frac{t^2}{\sigma^2}\right)\exp\left(-\frac{t^2}{2\sigma^2}\right).$$

- Gaussians, characterized by parameters μ and σ:

$$\phi(t; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right).$$

  Estimating Gaussian mixtures from sampled data is a much-studied problem in machine learning.

The key ingredient in solving these problems within the atomic norm framework is efficient (approximate) solution of the atom selection step. In some cases, this can be done in closed form, whereas for all the signals mentioned above, approximate solutions can be obtained via adaptive discretization.

C. OSCAR

The regularizer for the Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR) method is defined for $\boldsymbol{x} \in \mathbb{R}^p$ as follows:

$$\|\boldsymbol{x}\|_1 + c\sum_{j=1}^{p}\sum_{k=1}^{j}\max\{|\boldsymbol{x}_j|, |\boldsymbol{x}_k|\}$$

The atomic-norm formulation is obtained by defining the atoms to be the vectors with at most two non zero entries, each being ± 1. We considered the example of [17, Section 4, Example 5], corrupting the measurements with AWGN of standard deviation 0.05. CoGEnT was

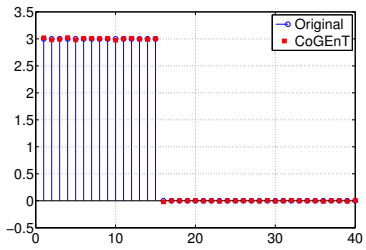


Fig. 8: Recovery of a vector with correlated variables obtained by applying CoGEnT to OSCAR

used to recover the reference vector β, varying the bound τ and choosing the value that performed best. Figure 8 shows that CoGEnT succeeds in recovering the solution.

VI. Reconstruction and Deconvolution

The deconvolution problem involves recovering a signal of the form $\boldsymbol{x} = \boldsymbol{x}^1 + \boldsymbol{x}^2$ from observations $\boldsymbol{y}$ via a sensing matrix $\boldsymbol{\Phi}$, where $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ can be expressed compactly with respect to different atomic sets $\mathcal{A}_1$ and $\mathcal{A}_2$. We mentioned several instances of such problems in Section I. Adopting the optimization-driven approach outlined in Section I, we arrive at the following convex optimization formulation:

$$\begin{aligned} \underset{\boldsymbol{x}^1, \boldsymbol{x}^2}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}(\boldsymbol{x}^1 + \boldsymbol{x}^2)\|^2 \\ \text{subject to} \quad & \|\boldsymbol{x}^1\|_{\mathcal{A}_1} \leq \tau_1 \text{ and } \|\boldsymbol{x}^2\|_{\mathcal{A}_2} \leq \tau_2. \end{aligned}$$

Algorithm 1 can be extended to this situation, as we describe informally now. Each iteration starts by choosing an atom from $\mathcal{A}_1$ that nearly minimizes its inner product with the gradient of the objective function with respect to $\boldsymbol{x}_1$; this is the forward step with respect to $\mathcal{A}_1$. One then performs a backward step for $\mathcal{A}_1$. Next follows a similar forward step with respect to $\mathcal{A}_2$, followed by a backward step for $\mathcal{A}_2$. We then proceed to the next iteration, unless convergence is flagged. Note that the backward steps are taken only if they do not deteriorate the objective function beyond a specified threshold. The entire procedure is repeated until a termination condition is satisfied.

In our first example, we consider the standard recovery of sparse + low rank matrices. We consider a matrix of size 50×50, which is a sum of a random rank 4 matrix and a sparse matrix with 100 entries. The sets $\mathcal{A}_1$ and $\mathcal{A}_2$ are defined in the usual way for these types of matrices. Figure 9 shows that CoGEnT recovers the components.

We consider now a novel application: *graph deconvolution*. To state this problem formally, consider two simple, undirected weighted graphs $\mathcal{G}_1 = (V, W_1)$ and $\mathcal{G}_2 = (V, W_2)$ where V represents a (common) vertex set and W_1, W_2 are the weighted adjacency matrices, with superposition $W = W_1 + W_2$. Problems of this form are of interest in *covariance estimation*: W_1 and W_2 may correspond to covariance matrices of random vectors X_1 and X_2, and from samples of $X = X_1 + X_2$, one may wish to recover the covariances W_1 and W_2.

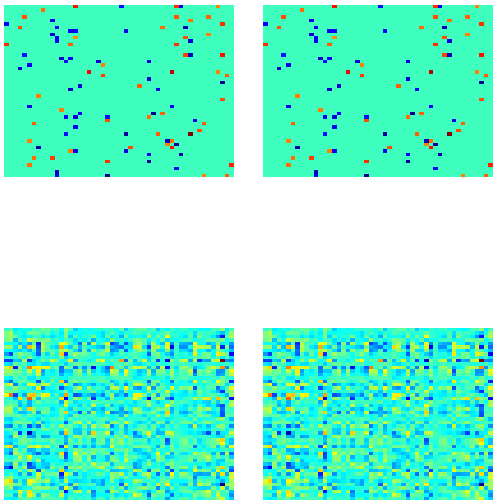


Fig. 9: Recovery of a sparse + low rank matrix. The left column shows true components, and the right column shows recovered components. The top row shows the sparse part and the bottom row shows the low-rank part. Error in each recovered component is at most 10^{-7}.

We consider a graph of $|V| = 50$ nodes in which $\mathcal{G}_1$ and $\mathcal{G}_2$ are each restricted to a specific family of graphs $\mathfrak{T}_1$ and $\mathfrak{T}_2$, respectively, with the following properties.

- $\mathfrak{T}_1$ is the class of all tree-structured graphs on 50 nodes. Note that the only information we exploit here is the fact that $\mathcal{G}_1$ is tree structured. Neither the edges of the tree nor the edge weights are known.
- $\mathfrak{T}_2$ is the class of two-dimensional 5×10 grid graphs on 50 nodes. The nodes of the graph are known up to a cyclic permutation. Once again, neither the edges of the graph nor the corresponding weights are known. The only information available is that one of the 50 cyclic permutations of the nodes yields the desired grid-structured graph.

For set $\mathfrak{T}_1$, we define the atomic set $\mathcal{A}_1$ to be the set of all matrices with Frobenius norm 1, whose nonzero

structure is the adjacency matrix of a tree.[2] For the set $\mathfrak{T}_2$ we define the atomic set $\mathcal{A}_2$ as follows. Let $\mathcal{P} \subseteq \mathbb{R}^{n\times n}$ denote the set of all permutation matrices corresponding to the cyclic permutations (that is, permutations in the cyclic group of order n). Let $\mathcal{G}(p,q)$ (with $pq = n$) denote the set of all weighted adjacency matrices (of unit Frobenius norm) of $p \times q$ grid graphs with a fixed canonical labeling of the nodes. The atomic set $\mathcal{A}_2$ is the set of weighted adjacency matrices for cyclic permutations of all these adjacency matrices.

Given these definitions, and assuming that we observe the full matrices, we state this deconvolution problem as:

$$\begin{array}{ll} \underset{X_1,X_2}{\text{minimize}} & \frac{1}{2}\|W - X_1 - X_2\|^2 \\ \text{subject to} & \|X_1\|_{\mathcal{A}_1} \leq \tau_1 \text{ and } \|X_2\|_{\mathcal{A}_2} \leq \tau_2. \end{array}$$

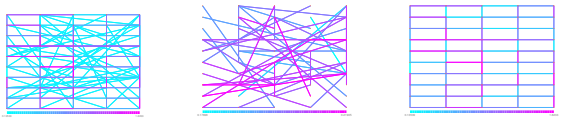
We need to compute the dual atomic norms to implement the forward steps in CoGEnT. The variational descriptions of the dual atomic norms are given by:

$$\|Y\|^*_{\mathcal{A}_i} = \max_{Z\in\mathcal{A}_i} \left[\text{trace}\,(ZY)\right]$$

For $\mathcal{A}_1$, the dual norm essentially amounts to computation of a maximum weight spanning tree, while for A_2, the dual norm can be computed in a straightforward way by sweeping through the n possible permutations of the grid graph to solve:

$$\|Y\|^*_{\mathcal{A}_2} = \max_{P\in\mathcal{P}, \|\mathcal{G}(p,q)\|_F \leq 1} \text{trace}\left(P^{'}\mathcal{G}(p,q)PY\right).$$

We implemented the deconvolution variant of CoGEnT with backward steps as described in Algorithm 2. Results are shown in Figure 10. CoGEnT achieves exact recovery; that is, the edges as well as the edge weights of the constituent graphs are correctly recovered.



(a) True signal is a superposition of a weighted tree and a weighted grid graph.

(b) Tree component, recovered by CoGEnT.

(c) Grid graph component, recovered by CoGEnT.

Fig. 10: Recovering constituent graph components from a superposition of weighted graphs. Edge weights are color-coded, with darker colors representing higher weights. CoGEnT correctly deconvolves the graph into its constituent components. (Best seen in color)

APPENDIX

Theorem III.2 is (except for a minor difference in the upper bounds on η) a true generalization of Theorem III.1, in that we recover the statement of Theorem III.1 by setting $\omega = 0$ in Theorem III.2. Likewise, the *proof* of Theorem III.1 can be obtained by setting $\omega = 0$ in Theorem III.2, so we prove only the latter result here.

A. Proof of Theorem III.2

Denote $f_t := f(\boldsymbol{x}_t)$, $\tilde{f}_t := f(\tilde{\boldsymbol{x}}_t)$, and $f_t^{FW} := f(\boldsymbol{x}_{t-1} + \gamma_t(\tau\boldsymbol{a}_t - \boldsymbol{x}_{t-1}))$. We have from the algorithm description that

$$f_{t+1} \leq \eta f_t + (1-\eta) f_{t+1}^{FW}.$$

[2]We learnt of the construction of tree-structured norms from James Saunderson, and express our gratitude for this insight.

For $\gamma \in [0,1]$, we define

$$\boldsymbol{x}_t(\gamma) := (1-\gamma)\boldsymbol{x}_t + \gamma\tau\boldsymbol{a}_{t+1}.$$

Because Step 6 of Algorithm 1 chooses the value of γ optimally, we have $f_{t+1}^{FW} = f(\boldsymbol{x}_t(\gamma_{t+1})) \leq f(\boldsymbol{x}_t(\gamma))$, for all $\gamma \in [0,1]$, and so

$$\begin{aligned}
& f_{t+1} \\
& \leq \eta f_t + (1-\eta) f_{t+1}^{FW} \\
& \leq \eta f_t + (1-\eta) f(\boldsymbol{x}_t(\gamma)) \\
& \leq \eta f_t + \\
& (1-\eta)\left[f_t + \nabla f(\boldsymbol{x}_t)^T(\boldsymbol{x}_t(\gamma) - \boldsymbol{x}_t)\right] + \\
& (1-\eta)\left[\frac{L}{2}\|\boldsymbol{x}_t(\gamma) - \boldsymbol{x}_t\|^2\right] \quad \text{(by definition of } L\text{)} \\
& = f_t + \\
& (1-\eta)\left[\nabla f(\boldsymbol{x}_t)^T\left((1-\gamma)\boldsymbol{x}_t + \gamma\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t\right)\right] + \\
& (1-\eta)\left[\frac{L}{2}\|(1-\gamma)\boldsymbol{x}_t + \gamma\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t\|^2\right] \\
& = f_t + (1-\eta)\left[\gamma\nabla f(\boldsymbol{x}_t)^T\left(\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t\right)\right] + \\
& (1-\eta)\left[\frac{L\gamma^2}{2}\|\tau\boldsymbol{a}_{t+1} - \boldsymbol{x}_t\|^2\right] \\
& \leq f_t + (1-\eta)\left[\gamma(1-\omega)\nabla f(\boldsymbol{x}_t)^T(\boldsymbol{x}^\star - \boldsymbol{x}_t)\right] + \\
& (1-\eta)\left[2\gamma^2 LR^2\tau^2\right] \quad \text{(see below)} \\
& \leq f_t + (1-\eta)\left[\gamma(1-\omega)(f_\star - f_t) + 2\gamma^2 LR^2\tau^2\right].
\end{aligned} \tag{14}$$

The last inequality follows from convexity of the objective function. The second-last inequality uses two results. First, note that the solution $\boldsymbol{x}^*$ can be expressed as follows:

$$\boldsymbol{x}^* = \sum_{\boldsymbol{a}\in\mathcal{A}} c^*_{\boldsymbol{a}}\boldsymbol{a}, \quad \text{for } c^* \geq 0 \text{ with } \textstyle\sum_{\boldsymbol{a}\in\mathcal{A}} c^*_{\boldsymbol{a}} \leq \tau.$$

We therefore have

$$\begin{aligned}
&\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}^* - \boldsymbol{x}_t \rangle \\
&= \left\langle \nabla f(\boldsymbol{x}_t), \left(\sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}}^* \boldsymbol{a} \right) - \boldsymbol{x}_t \right\rangle \\
&\geq \left(\sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}}^* \right) \min_{\boldsymbol{a} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{a} \rangle - \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t \rangle \\
&\geq \min_{\boldsymbol{a} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t), \tau \boldsymbol{a} - \boldsymbol{x}_t \rangle \\
&\geq \frac{1}{1-\omega} \langle \nabla f(\boldsymbol{x}_t), \tau \boldsymbol{a}_{t+1} - \boldsymbol{x}_t \rangle,
\end{aligned}$$

by the definition of $\boldsymbol{a}_{t+1}$ in (11) and noting that $\min_{\boldsymbol{a} \in \mathcal{A}} \langle \nabla f(\boldsymbol{x}_t) \boldsymbol{a} \rangle \leq 0$. Second, we use the definition of R together with $\|\boldsymbol{x}_t\|_{\mathcal{A}} \leq \tau$ and $\boldsymbol{a}_{t+1} \in \mathcal{A}$ to deduce

$$\|\tau \boldsymbol{a}_{t+1} - \boldsymbol{x}_t\| \leq \tau \left(\|\boldsymbol{a}_{t+1}\| + \|\boldsymbol{x}_t / \tau\| \right) \leq 2\tau R,$$

which we can use to bound the squared-norm term. By subtracting f^* from both sides of (14), and defining

$$\delta_t := f(\boldsymbol{x}_t) - f^*, \tag{15}$$

we obtain that

$$\delta_{t+1} \leq \left[1 - \gamma(1-\eta)(1-\omega) \right] \delta_t + 2(1-\eta) L R^2 \gamma^2 \tau^2, \tag{16}$$

for all $\gamma \in [0, 1]$. This inequality implies immediately that $\{\delta_t\}_{t=0,1,2,\ldots}$ is a decreasing sequence, since $\gamma = 0$ is always a valid choice in (16).

Note that $\delta_0 = f_0 - f_\star = D$. For the first iteration $t = 0$, set $\gamma = 1$ in (16) to obtain a further bound on δ_1:

$$\delta_1 \leq [\eta + \omega(1-\eta)] D + 2(1-\eta) L R^2 \tau^2 = \tilde{C}_1.$$

For subsequent iterations $t \geq 1$, we consider the following choice of γ:

$$\tilde{\gamma}_t := \frac{\delta_t}{2\tilde{C}_1}.$$

By monotonicity of $\{\delta_t\}$ and the bound above on δ_1, we have $\tilde{\gamma}_t \leq 1/2$ for all $t \geq 1$. By substituting the choice $\gamma = \tilde{\gamma}_t$ into (16), we obtain

$$\begin{aligned}
\delta_{t+1} &\leq \delta_t - \delta_t^2 \frac{(1-\eta)(1-\omega)\tilde{C}_1 - (1-\eta) L R^2 \tau^2}{2\tilde{C}_1^2} \\
&= \delta_t - \frac{\delta_t^2}{\tilde{C}}.
\end{aligned} \tag{17}$$

The denominator of $\tilde{C}$ is positive because $\eta \in (0, 1/3]$ and $\omega \in (0, 1/4]$ together imply that

$$(1-\omega)\tilde{C}_1 - L R^2 \tau^2 > 2(1-\omega)(1-\eta) L R^2 \tau^2 - L R^2 \tau^2 \geq 0.$$

Note too that

$$\tilde{C} = \frac{2\tilde{C}_1^2}{(1-\eta)((1-\omega)\tilde{C}_1 - L R^2 \tau^2)} > 2\tilde{C}_1,$$

so that $\delta_1 \leq \tilde{C}/2$. An argument from [44, Lemma 2.1] yields the result. Since $\delta_1 \leq \tilde{C}/2$, the bound (12) holds for $t = 1$. Since $\{\delta_t\}$ is a decreasing sequence, we have $\delta_t \leq \tilde{C}/2$ for all $t \geq 1$. For the inductive step, assume that (12) holds for some $t \geq 1$. Since the right-hand side of (17) is an increasing function of δ_t for all $\delta_t \in (0, \tilde{C}/2)$, this quantity can be upper-bounded by substituting the upper bound $\tilde{C}/(t+1)$ for δ_t, to obtain

$$\begin{aligned}
\delta_{t+1} &\leq \delta_t - \frac{\delta_t^2}{\tilde{C}} \leq \frac{\tilde{C}}{(t+1)} - \frac{\tilde{C}}{(t+1)^2} \\
&= \frac{\tilde{C} t}{(t+1)^2} = \frac{\tilde{C} t (t+2)}{(t+1)^2 (t+2)} \leq \frac{\tilde{C}}{t+2},
\end{aligned}$$

establishing the inductive step and completing the proof.

REFERENCES

[1] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[3] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

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

[5] N. S. Rao, R. D. Nowak, S. J. Wright, and N. G. Kingsbury, "Convex approaches to model wavelet sparsity patterns," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1917–1920.

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

[7] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 433–440.

[8] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Convex and network flow optimization for structured sparsity," *The Journal of Machine Learning Research*, vol. 12, pp. 2681–2720, 2011.

[9] S. Negahban and M. J. Wainwright, "Joint support recovery under high-dimensional scaling: Benefits and perils of l1-regularization," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1161–1168, 2008.

[10] B. A. Turlach, W. N. Venables, and S. J. Wright, "Simultaneous variable selection," *Technometrics*, vol. 47, no. 3, pp. 349–363, 2005.

[11] N. S. Rao, B. Recht, and R. D. Nowak, "Universal measurement bounds for structured sparse signal recovery," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 942–950.

[12] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressive sensing off the grid," in *50th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2012, pp. 778–785.

[13] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," in *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*. IEEE, 2010, pp. 1913–1917.

[14] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2010.

[15] S. Chatterjee, A. Banerjee, S. Chatterjee, and A. R. Ganguly, "Sparse group lasso for regression on land climate variables." in *ICDM Workshops*, 2011, pp. 1–8.

[16] N. Rao, C. Cox, R. Nowak, and T. T. Rogers, "Sparse overlapping sets lasso for multitask learning and its application to fmri analysis," in *Advances in Neural Information Processing Systems*, 2013, pp. 2202–2210.

[17] H. D. Bondell and B. J. Reich, "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar," *Biometrics*, vol. 64, no. 1, pp. 115–123, 2008.

[18] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," preprint arXiv:1210.7559, 2012.

[19] M. B. McCoy, V. Cevher, Q. T. Dinh, A. Asaei, and L. Baldassarre, "Convexity in source separation: Models, geometry, and algorithms," preprint arXiv:1311.0258, 2013.

[20] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements." in *NIPS*, 2011, pp. 1089–1097.

[21] A. Jalali, P. D. Ravikumar, S. Sanghavi, and C. Ruan, "A dirty model for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 23, pp. 964–972, 2010.

[22] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[23] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[24] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

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

[26] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[27] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 427–435.

[28] J. C. Dunn, "Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals," *SIAM Journal on Control and Optimization*, vol. 17, no. 2, pp. 187–211, 1979.

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

[30] M. Dudik, Z. Harchaoui, and J. Malick, "Learning with matrix gauge regularizers," *NIPS Optimization Workshop*, 2011.

[31] R. M. Freund and P. Grigas, "New analysis and results for the conditional gradient method," preprint arXiv:1307.0873, 2013.

[32] 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.

[33] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[34] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

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

[36] Y. Nesterov, "Gradient methods for minimizing composite objective functions," *Mathematical Programming, Series B*, 2013, to appear.

[37] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[38] S. Lacoste-Julien, M. Jaggi, M. Schmidt, P. Pletscher *et al.*, "Block-coordinate Frank-Wolfe optimization for structural SVMs," *International Conference on Machine Learning*, pp. 53–61, 2013.

[39] J. Guelat and P. Marcotte, "Some comments on wolfe's away step," *Mathematical Programming*, vol. 35, no. 1, pp. 110–119, 1986.

[40] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4689–4708, 2011.

[41] P. Jain, A. Tewari, and I. S. Dhillon, "Orthogonal matching pursuit with replacement," *Advances in Neural Information Processing Systems*, pp. 1215–1223, 2011.

[42] C. C. Johnson, A. Jalali, and P. D. Ravikumar, "High-dimensional sparse inverse covariance estimation using greedy methods," *International Conference on Artificial Intelligence and Statistics*, pp. 574–582, 2012.

[43] J. Liu, R. Fujimaki, and J. Ye, "Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint," preprint arXiv:1401.0086, 2013.

[44] A. Beck and M. Teboulle, "A conditional gradient method with linear rate of convergence for solving convex linear systems," *Mathematical Methods of Operations Research*, vol. 59, no. 2, pp. 235–247, 2004.

[45] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[46] B. Dumitrescu, *Positive Trigonometric Polynomials and Signal Processing Applications*. Springer, 2007.