SIAM J. Optim.
Vol. 27, No. 2, pp. 616–639

# THE ALTERNATING DESCENT CONDITIONAL GRADIENT METHOD FOR SPARSE INVERSE PROBLEMS[*]

NICHOLAS BOYD[†], GEOFFREY SCHIEBINGER[†], AND BENJAMIN RECHT[‡]

**Abstract.** We propose a variant of the classical conditional gradient method for sparse inverse problems with differentiable observation models. Such models arise in many practical problems including superresolution microscopy, time-series modeling, and matrix completion. Our algorithm combines nonconvex and convex optimization techniques: we propose global conditional gradient steps alternating with nonconvex local search exploiting the differentiable observation model. This hybridization gives the theoretical global optimality guarantees and stopping conditions of convex optimization along with the performance and modeling flexibility associated with nonconvex optimization. Our experiments demonstrate that our technique achieves state-of-the-art results in several applications.

**Key words.** conditional gradient method, compressed sensing, measures, sparsity, inverse problems, convex optimization, semi-infinite programming

**AMS subject classifications.** 90C25, 90C30, 90C34, 90C49, 90C52, 90C90, 49M37, 65K05

**DOI.** 10.1137/15M1035793

## 1. Introduction.

A ubiquitous prior in modern statistical signal processing asserts that an observed signal is the noisy observation of a few weighted sources. In other words, compared to the entire dictionary of possible sources, the set of sources actually present is *sparse*. In the most abstract formulation of this prior, each source is chosen from a nonparametric dictionary, but in many cases of practical interest the sources are parameterized. Hence, solving the sparse inverse problem amounts to finding a collection of a few parameters and weights that adequately explains the observed signal.

As a concrete example, consider the idealized task of identifying the aircraft that lead to an observed radar signal. The sources are the aircraft themselves, and each is parameterized by, perhaps, its position and velocity relative to the radar detector. The sparse inverse problem is to recover the number of aircraft present, along with each of their parameters.

Any collection of weighted sources can be represented as a measure on the parameter space: each source corresponds to a single point mass at its corresponding parameter value. We will call atomic measures supported on very few points *sparse* measures. When the parameter spaces are infinite—for example, the set of all velocities

[*] Received by the editors August 21, 2015; accepted for publication (in revised form) January 9, 2017; published electronically April 13, 2017.
http://www.siam.org/journals/siopt/27-2/M103579.html

**Funding:** The work of the third author was generously supported by ONR awards N00014-11-1-0723 and N00014-13-1-0129, NSF awards CCF-1148243 and CCF-1217058, AFOSR award FA9550-13-1-0138, and a Sloan Research Fellowship. The work of the second author was generously supported by NSF award CCF-1148243. The work of the first author was generously supported by a Google Fellowship from the Hertz Foundation. This research is supported in part by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Adatao, Adobe, Apple, Inc., Blue Goji, Bosch, C3Energy, Cisco, Cray, Cloudera, EMC2, Ericsson, Facebook, Guavus, HP, Huawei, Informatica, Intel, Microsoft, NetApp, Pivotal, Samsung, Schlumberger, Splunk, Virdata, and VMware.

[†] Statistics, UC Berkeley, Berkeley, CA 94704 (nickboyd@berkeley.edu, geoff@stat.berkeley.edu).

[‡] Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720 (brecht@berkeley.edu).

and positions of aircraft—the space of bounded measures over such parameters is infinite dimensional. This means that optimization problems searching for parsimonious explanations of the observed signal must operate over an infinite-dimensional space.

Many alternative formulations of the sparse inverse problem have been proposed to avoid the infinite-dimensional optimization required in the sparse measure setup. The most canonical and widely applicable approach is to form a discrete grid over the parameter space and restrict the search to measures supported on the grid. This restriction produces a finite-dimensional optimization problem [55, 38, 52]. In certain special cases, the infinite-dimensional optimization problem overmeasures can be reduced to a problem of moment estimation, and spectral techniques or semidefinite programming can be employed [24, 14, 53, 10]. More recently, in light of much of the work on compressed sensing and its generalizations, another proposal operates on atomic norms over data [12], opening other algorithmic possibilities.

While these finite-dimensional formulations are appealing, they all essentially treat the space of sources as an unstructured set, ignoring natural structure (such as differentiability) present in many applications. All three of these techniques have their individual drawbacks, as well. Gridding only works for very small parameter spaces, and introduces artifacts that often require heuristic postprocessing [52]. Moment methods have limited applicability, are typically computationally expensive, and, moreover, are sensitive to noise and estimates of the number of sources. Finally, atomic-norm techniques do not recover the parameters of the underlying signal, and as such are more naturally applied to denoising problems. For a much more in-depth discussion of atomic norms, see Appendix A.

In this paper, we argue that all of these issues can be alleviated by returning to the original formulation of the estimation problem as an optimization problem over a space of measures. Working with measures explicitly exposes the underlying parameter space, which allows us to consider algorithms that make local moves within parameter space. We demonstrate that operating on an infinite-dimensional space of measures is not only feasible algorithmically, but that the resulting algorithms outperform techniques based on gridding or moments on a variety of real-world signal processing tasks. We formalize a general approach to solving parametric sparse inverse problems via the conditional gradient method (CGM), also know as the Frank–Wolfe algorithm. In section 3, we show how to augment the classical CGM with nonconvex local search exploiting structure in the parameter space. This hybrid scheme, which we call the alternating descent conditional gradient method (ADCG), enjoys both the rapid local convergence of nonconvex programming algorithms and the stability and global convergence guarantees associated with convex optimization. The theoretical guarantees are detailed in Appendix B, where we bound the convergence rate of our algorithm and also guarantee that it can be run with bounded memory. Moreover, in section 5 we demonstrate that our approach achieves state-of-the-art performance on a diverse set of examples.

### 1.1. Mathematical setup.

In this subsection we formalize the sparse inverse problem as an optimization problem over measures and discuss a convex heuristic.

We assume the existence of an underlying collection of objects, called sources. Each source has a scalar weight $w$, and a parameter $\theta \in \Theta$. We require the parameter space be measurable (that is, come equipped with a $\sigma$-algebra) and amenable to local, derivative-based optimization; formally, we need $\Theta$ to be a compact subset of a differentiable manifold. Some examples to keep in mind would be $\Theta$ as a compact subset of $\mathbb{R}^p$ for some small $p$, or the sphere $\mathcal{S}^p$ considered as a differentiable manifold.

An element $\theta$ of the parameter space $\Theta$ may describe, for instance, the position, orientation, and polarization of a source. The weight $w$ may encode the intensity of a source, or the distance of a source from the observation device. Our goal is to recover the number of sources present, along with their individual weights and parameters. We do not observe the sources directly, but instead are given a single, noisy observation in $\mathbb{R}^d$.

The observation model we use is completely specified by a function $\psi : \Theta \to \mathbb{R}^d$, which gives the $d$-dimensional observation of a single, unit-weight source parameterized by a point in $\Theta$. A single source with parameter $\theta$ and weight $w$ generates the observation $w\psi(\theta) \in \mathbb{R}^d$: that is, the observation of a lone source is homogeneous of degree one in its weight. Finally, we assume that the observation generated by a weighted collection of sources is additive. In other words, the (noise-free) observation of a weighted collection is generated by the mapping

$$\{(w_i, \theta_i)\}_{i=1}^K \mapsto \sum_{i=1}^K w_i \psi(\theta_i) \in \mathbb{R}^d. \tag{1.1}$$

We refer to the collection $\{(w_i, \theta_i)\}_{i=1}^K$ as the *signal parameters*, and the vector $\sum_{i=1}^K w_i \psi(\theta_i) \in \mathbb{R}^d$ as the noise-free *observation*. We require that $\psi$ be bounded: $\|\psi(\theta)\|_2^2 \leq 1$ for all $\theta$, and further that $\psi$ be *differentiable* in $\theta$. Finally, let us emphasize that we make no further assumptions about $\psi$: in particular, it does *not* need to be linear. It's worth noting here that with $K$ and $\theta_1, \dots, \theta_K$ held fixed, (1.1) is *linear* in the weights $w_1, \dots, w_K$.

Our goal is to recover the true weighted collection of sources, $\{(\tilde{w}_i, \tilde{\theta}_i)\}_{i=1}^{\tilde{K}}$, from a single noisy observation:

$$y = \sum_{i=1}^{\tilde{K}} \tilde{w}_i \psi(\tilde{\theta}_i) + \nu.$$

Here $\nu$ is an additive noise term. We emphasize that the goal is *not* to denoise the vector $y$: that is, we are not satisfied with recovering the noise-free observation (i.e., the vector $\sum_{i=1}^{\tilde{K}} \tilde{w}_i \psi(\tilde{\theta}_i)) \in \mathbb{R}^d$), but rather we require an estimate of the true signal parameters $\{(\tilde{w}_i, \tilde{\theta}_i)\}$. This is in stark contrast with the atomic-norm case discussed in Appendix A.

One approach would be to attempt to minimize a (differentiable) convex loss, $\ell$, of the residual between the observed vector $y$ and the expected output for an estimated collection of sources:

$$\begin{array}{ll} \underset{w,\theta,K}{\text{minimize}} & \ell\left(\sum_{i=1}^K w_i \psi(\theta_i) - y\right) \\ \text{subject to} & K \leq N. \end{array} \tag{1.2}$$

Here $N$ is a posited upper bound on the number of sources. For example, when $\ell$ is the negative log-likelihood of the noise term $\nu$, problem (1.2) corresponds to maximum-likelihood estimation of the true sources. Unfortunately, (1.2) is nonconvex in the variables $w$, $\theta$, and $K$. As such, algorithms designed to solve this problem are hard to reason about and come with few guarantees. Also, in practice they often suffer from sensitivity to initialization. In this paper, we *lift* the problem to a space of (signed) measures on $\Theta$; this lifting allows us to apply a natural heuristic to devise a convex surrogate for problem (1.2).

We can encode an arbitrary, weighted collection of sources as an atomic measure $\mu$ on $\Theta$, with mass $w_i$ at point $\theta_i$: $\mu = \sum_{i=1}^{K} w_i \delta_{\theta_i}$. As a consequence of the additivity and homogeneity in our observation model, the total observation of a collection of sources encoded in the measure $\mu$ is a linear function $\Phi$ of $\mu$:

$$\Phi\mu = \int \psi(\theta) d\mu(\theta).$$

We call $\Phi$ the *forward operator*. For atomic measures of the form $\mu = \sum_{i=1}^{n} w_i \delta_{\theta_i}$, this clearly agrees with (1.1); but it is defined for all measures on $\Theta$.

We now introduce the sparse inverse problem as an optimization problem over the Banach space of bounded, signed measures on the measurable space $\Theta$ equipped with the total variation norm [17]. To reiterate, our goal is to recover $\mu_{\text{true}}$ from an observation

$$y = \Phi\mu_{\text{true}} + \nu$$

corrupted by the noise term, $\nu$. Recovering the signal parameters without any prior information is, in most interesting problems, impossible; the operator $\Phi$ is almost never injective. However, in a sparse inverse problem we have the prior belief that the number of sources present, while still unknown, is small. That is, we assume that $\mu_{\text{true}}$ is an atomic measure supported on very few points.

To make the connection to compressed sensing clear, we refer to such measures as *sparse* measures. Note that while we are using the language of *recovery* or *estimation* in this section, the optimization problem we introduce is also applicable in cases where these may not be a true measure underlying the observation model. In section 2 we give several examples that are not recovery problems.

We estimate the signal parameters encoded in $\mu_{\text{true}}$ by minimizing the loss $\ell$ of the residual between $y$ and $\Phi\mu$:

$$\begin{array}{ll} \text{minimize} & \ell\left(\Phi\mu - y\right) \\ \text{subject to} & |\text{supp}(\mu)| \leq N, \end{array} \tag{1.3}$$

where the optimization is over the Banach space of signed measures (on $\Theta$) equipped with the total variation norm. Here we constrain the cardinality of the support of the measure $\mu$ by $N$, a posited upper bound on the size of the support of the true measure $\mu_{\text{true}}$. Although here and elsewhere in the paper we place no constraint on the sign of $w$ (and hence $\mu$), all of our discussion and algorithms can be easily extended to the nonnegative case by adding the requirement that $\mu$ be a nonnegative measure.

While the objective function in (1.3) is convex, the constraint on the support of $\mu$ is nonconvex. A common heuristic in this situation is to replace the nonconvex constraint with a convex surrogate. The standard surrogate for a cardinality constraint on a measure is a constraint on the total variation [10]. This substitution results in the following convex approximation to (1.3):

$$\begin{array}{ll} \text{minimize} & \ell\left(\Phi\mu - y\right) \\ \text{subject to} & |\mu|(\Theta) \leq \tau. \end{array} \tag{1.4}$$

Here $\tau > 0$ is a parameter that controls the total mass of $\mu$ and empirically controls the cardinality of solutions to (1.4). While problem (1.4) is convex, it is over an infinite-dimensional space, and it is not possible to represent an arbitrary measure in a computer. A priori, an approximate solution to (1.4) may have arbitrarily large support, though we prove in section B that we can always find solutions supported on at most $d+1$ points. In practice, however, we are interested in approximate solutions of (1.4) supported on far, far fewer than $d+1$ points.

In this paper, we propose an algorithm to solve (1.4) in the case where $\Theta$ is amenable to local, derivative-based optimization—in other words, the case where $\psi$ is differentiable. Our algorithm is based on a variant of the CGM that takes advantage of the differentiable nature of $\psi$, and is guaranteed to produce approximate solutions with bounded support.

*Relationship to the lasso.* Readers familiar with techniques for estimating sparse vectors may recognize (1.4) as a continuous analogue of the standard lasso. In particular, the standard lasso is an instance of (1.4) with $\ell(r) = \frac{1}{2}\|r\|_2^2$ and $\Theta = \{1, \ldots, k\}$. In that case, a measure over $\Theta$ can be represented as a vector $v$ in $\mathbb{R}^k$ and the forward operator $\Phi$ as a matrix in $\mathbb{R}^{d \times k}$. The total variation of the measure $v$ is then simply $\sum_i |v_i| = \|v\|_1$. We caution the reader that this discrete setup is substantially different as the parameter space has no differential structure. However, to make the connection to the finite-dimensional case clear, we will use the notation $\|\mu\|_1$ to refer to the total variation of the measure $\mu$.

*A note on measures.* While the optimization problem (1.4) has as the decision variable a general measure $\mu$ on $\Theta$, due to the nature of the algorithms we discuss we will only ever deal with sparse measures. Sparse measures, that is measures of the form $\mu = \sum_{i=1}^K w_i \psi(\theta_i)$, can always be thought of as simple sets of weighted parameters: $\{(w_i, \theta_i)\}_{i=1}^K$ or, equivalently a pair of vectors $w \in \mathbb{R}^K$, $\vec{\theta} \in \Theta^K$. Indeed, we will often move back and forth between these equivalent representations. The reader unfamiliar with measure theory can think of the algorithm as operating on this alternative representation directly. Two important identities to keep in mind when dealing with sparse measures are (1.1) and

$$\|\mu\|_1 = \|w\|_1.$$

For a review of basic measure theory, see [22].

## 2. Example applications.

Many practical problems can be formulated as instances of (1.4). In this section we briefly outline a few examples to motivate our study of this problem.

*Superresolution imaging.* The diffraction of light imposes a physical limit on the resolution of optical images. The goal of superresolution imaging is to remove the blur induced by diffraction as well as the effects of pixelization and noise. For images composed of a collection of point sources of light, this can be posed as a sparse inverse problem as follows. The parameters $\theta_1, \ldots, \theta_K$ denote the locations of $K$ point sources (in $[0,1]^2$ or $[0,1]^3$), and $w_i > 0$ denotes the intensity, or brightness, of the $i$th source. The image of the $i$th source is given by $w_i \psi(\theta_i)$, where $\psi$ is the pixelated point spread function of the imaging apparatus.

By solving a version of (1.4) it is sometimes possible to localize the point sources better than the diffraction limit—even with extreme pixelization. Astronomers use this framework to deconvolve images of stars to angular resolution below the Rayleigh limit [42]. In biology this tool has revolutionized imaging of subcellular features [20, 47]. A variant of this framework allows imaging through scattering media [37]. In section 5.1, we show that our algorithm improves upon the current state of the art for localizing point sources in a fluorescence microscopy challenge dataset.

*Linear system identification.* Linear time-invariant (LTI) dynamical systems are used to model many physical systems. Such a model describes the evolution of an output $y_t \in \mathbb{R}$ based on the input $u_t \in \mathbb{R}$, where $t \in \mathbb{Z}_+$ indexes time. The internal state at time $t$ of the system is parameterized by a vector $x_t \in \mathbb{R}^m$, and its relationship to the output is described by

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t. \end{aligned}$$

Here $C$ is a fixed matrix, while $x_0, A$, and $B$ are unknown parameters.

Linear system identification is the task of learning these unknown parameters from input-output data—that is a sequence of inputs $u_1, \ldots, u_T$ and the observed sequence of outputs $y_1, \ldots, y_T$ [49, 24]. We pose this task as a sparse inverse problem. Each source is a small LTI system with 2-dimensional state—the measurement model gives the output of the small system on the given input. To be concrete, the parameter space $\Theta$ is all tuples of the form $(x_0, r, \alpha, B)$ where $x_0$ and $B$ both lie in the $\ell_\infty$ unit ball in $\mathbb{R}^2$, $r$ is in $[0, 1]$, and $\alpha$ is in $[0, \pi]$. The LTI system that each source describes has

$$A = r \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The mapping $\psi$ from the parameters $(x_0, r, \alpha, B)$ to the output of the corresponding LTI system on input $u_1, \ldots, u_T$ is differentiable. In terms of the overall LTI system, adding the output of two weighted sources corresponds to concatenating the corresponding parameters.

In section 5.1, we show that our algorithm matches the state of the art on two standard system identification datasets.

*Matrix completion.* The task of matrix completion is to estimate all entries of a large matrix given observations of a few entries. Clearly this task is impossible without prior information or assumptions about the matrix. If we believe that a low-rank matrix will approximate the truth well, a common heuristic is to minimize the squared error subject to a nuclear norm bound. For background in the theory and practice of matrix completion under this assumption, see [4, 11]. We solve the following optimization problem

$$\min_{\|A\|_* \leq \tau} \|M(A) - y\|^2.$$

Here $M$ is the masking operator, that is, the linear operator that maps a matrix $A \in \mathbb{R}^{n \times m}$ to the vector containing its observed entries, and $y$ is the vector of observed entries. We can rephrase this in our notation by letting

$$\Theta = \{(u, v) \in \mathbb{R}^n \times \mathbb{R}^m : \|u\|_2 = \|v\|_2 = 1\}, \quad \psi((u, v)) = M(uv^T),$$

and $\ell(\cdot) = \|\cdot\|^2$. In section 5.1, we show that our algorithm achieves state-of-the-art results on the Netflix Challenge, a standard benchmark in matrix completion.

*Bayesian experimental design.* In experimental design we seek to estimate a vector $x \in \mathbb{R}^d$ from measurements of the form

$$y_i = f(\theta_i)^T x + \epsilon_i.$$

Here $f : \Theta \to \mathbb{R}^d$ is a known differentiable feature function and $\epsilon_i$ are independent noise terms. We want to choose $\theta_i, \ldots, \theta_k$ to minimize our uncertainty about $x$—if each measurement requires a costly experiment, this corresponds to getting the most information from a fixed number of experiments. For background, see [41].

In general, this task in intractable. However, if we assume the $\epsilon_i$ are independently distributed as standard normals and $x$ comes from a standard normal prior we can analytically derive the posterior distribution of $x$ given $y_1, \dots, y_k$, as the full joint distribution of $x, y_1, \dots, y_k$ is normal.

One notion of how much information $y_1, \dots, y_k$ carry about $x$ is the entropy of the posterior distribution of $x$ given the measurements. We can then choose $\theta_1, \dots, \theta_k$ to minimize the entropy of the posterior, which is equivalent to minimizing the (log) volume of an uncertainty ellipsoid. With this setup, the posterior entropy is (up to additive constants and a positive multiplicative factor) simply

$$-\log\det\left(I + \sum_i f(\theta_i) f(\theta_i)^T\right)^{-1}.$$

To put this in our framework, we can take $\psi(\theta) = f(\theta)f(\theta)^T$, $y = 0$, and $\ell(M) = -\log\det(I + M)^{-1}$. We relax the requirement to choose exactly $k$ measurement parameters and instead search for a nonnegative sparse measure with bounded total mass, giving us an instance of (1.4).

*Fitting mixture models to data.* Given a parametric distribution $P(x|\theta)$ we consider the task of recovering the components of a mixture model from independently and identically distributed (i.i.d.) samples. For background, see [34]. To be more precise, we are given data $\{x_1, \dots, x_d\}$ sampled i.i.d. from a distribution of the form $P(x) = \int_{\theta\in\Theta} P(x|\theta)\pi(\theta)$. The task is to recover the mixing distribution $\pi$. If we assume $\pi$ is sparse, we can phrase this as a sparse inverse problem. To do so, we choose $\psi(\theta) = (P(x_i|\theta))_{i=1}^d$. A common choice for $\ell$ is the (negative) log-likelihood of the data, i.e., $y = 0$, $\ell(p) = -\sum_i \log p_i$. The obvious constraints here are $\int d\pi(\theta) \le 1, \pi \ge 0$.

*Design of numerical quadrature rules.* In many numerical computing applications we require fast procedures to approximate integration against a fixed measure. One way to do this is use a quadrature rule:

$$\int f(\theta) dp(\theta) \simeq \sum_{i=1}^k w_i f(x_i).$$

The quadrature rule, given by $w_i \in \mathbb{R}$ and $x_i \in \Theta$, is chosen so that the above approximation holds for functions $f$ in a certain function class. The pairs $(x_i, w_i)$ are known as quadrature nodes. In practice, we want quadrature rules with very few nodes to speed evaluation of the rule.

Often we don't have an a priori description of the function class from which $f$ is chosen, but we might have a finite number of examples of functions in the class, $f_1, \dots, f_d$, along with their integrals against $p$, $y_1, \dots, y_d$. In other words, we know that

$$\int f_i(\theta) dp(\theta) = y_i.$$

A reasonable quadrature rule should approximate the integrals of the known $f_i$ well.

We can phrase this task as a sparse inverse problem where each source is a single quadrature node. In our notation, $\psi(\theta) = (f_1(\theta), \dots, f_d(\theta))$. Assuming each function $f_i$ is differentiable, $\psi$ is differentiable. A common choose of $\ell$ for this application is simply the squared loss. For more discussion of the design of quadrature rules using the CGM, see [5, 35].

*Neural spike sorting.* In this example we consider the voltage $v$ recorded by an extracellular electrode implanted in the vicinity of a population of neurons. Suppose that this population of neurons contains $T$ types of neurons, and that when a neuron of type $k$ fires at time $t \in \mathbb{R}$, an action potential of the form $\psi(t, k)$ is recorded. Here $\psi : [0, 1] \times \{1, \ldots, T\} \to \mathbb{R}^d$ is a vector of voltage samples. Note that $\psi$ is not differentiable in its last argument. The algorithms we discuss in this paper can still be applied in this case, but any steps leveraging differentiability of $\psi$ must operate only on its first argument. If we denote the parameters of the $i$th neuron by $\theta_i = (t_i, k_i)$, then the total voltage $v \in \mathbb{R}^d$ can be modeled as a superposition of these action potentials:

$$v = \sum_{i=1}^{K} w_i \psi(\theta_i).$$

Here the weights $w_i > 0$ can encode the distance between the $i$th neuron and the electrode. The sparse inverse problem in this application is to recover the parameters $\theta_1, \ldots, \theta_K$ and weights $w_1, \ldots, w_K$ from the voltage signal $v$. For background, see [19].

*Designing radiation therapy.* External radiation therapy is a common treatment for cancer in which several beams of radiation are fired at the patient to irradiate tumors. The collection of beam parameters (their intensities, positions, and angles) is called the treatment plan, and is chosen to minimize an objective function specified by an oncologist. The objective usually rewards giving large doses of radiation to tumors, and low dosages to surrounding healthy tissue and vital organs. Plans with few beams are desired as repositioning the emitter takes time—increasing the cost of the procedure and the likelihood that the patient moves enough to invalidate the plan.

A beam fired with intensity $w > 0$ and parameter $\theta$ delivers a radiation dosage $w\psi(\theta) \in \mathbb{R}^d$. Here the output is interpreted as the radiation delivered to each of $d$ voxels in the body of a patient. The radiation dosage from beams with parameters $\theta_1, \ldots, \theta_K$ and intensities $w_1, \ldots, w_K$ add linearly, and the objective function is convex. For background, see [28].

## 3. Conditional gradient method.

In this section we present our main algorithmic development. We begin with a review of the classical CGM for finite-dimensional convex programs. We then apply the CGM to the sparse inverse problem (1.4). In particular, we augment this algorithm with a local search subroutine that significantly improves the practical performance of the CGM.

The classical CGM solves the following optimization problem:

$$\underset{x \in \mathcal{C}}{\text{minimize}}\, f(x), \tag{3.1}$$

where $\mathcal{C}$ is a bounded convex set and $f$ is a differentiable convex function.

The CGM proceeds by iteratively solving linearized versions of (3.1). At iteration $k$, we form the standard linear approximation to the function $f$ at the current point $x_k$:

$$\hat{f}_k(s) = f(x_k) + f'(s - x_k; x_k).$$

Here $f'(s - x_k; x_k)$ is the directional derivative of the function $f$ at $x_k$ in the direction $s - x_k$. When $f$ is differentiable, $f'(s - x_k; x_k)$ is more commonly written as $\langle \nabla f(x_k), s - x_k \rangle$; we use the directional derivative to make the extension to optimization on measures easier. As $f$ is convex, this approximation is a global lower bound. We then minimize the linearization over the feasible set to get a potential solution $s_k$. As $s_k$ minimizes a simple approximation of $f$ that degrades with distance from $x_k$ we take a convex combination of $s_k$ and $x_k$ as the next iterate. We summarize this method in Algorithm 1.

**Algorithm 1** Conditional gradient method (CGM).

```
For $k = 1, \dots k_{\max}$
    1. Linearize: $\hat{f}_k(s) \leftarrow f(x_k) + f'(s - x_k; x_k)$.
    2. Minimize: $s_k \ni \arg\min_{s \in \mathcal{C}} \hat{f}_k(s)$.
    3. Tentative update: $\tilde{x}_{k+1} \leftarrow \frac{k}{k+2} x_k + \frac{2}{k+2} s_k$.
    4. Final update: Choose $x_{k+1}$ such that $f(x_{k+1}) \leq f(\tilde{x}_{k+1})$.
```

It is important to note that minimizing $\hat{f}_k(s)$ over the feasible set $\mathcal{C}$ in step 2 may be quite difficult and requires an application-specific subroutine.

One of the more remarkable features of the CGM is step 4. While the algorithm converges using only the tentative update in step 3, all of the convergence guarantees of the algorithm are preserved if one replaces $\tilde{x}_{k+1}$ with *any* feasible $x_{k+1}$ that achieves a smaller value of the objective. There are thus many possible choices for the final update in step 4, and the empirical behavior of the algorithm can be quite different for different choices. One common modification is to do a line search:

$$x_{k+1} = \underset{x \in \operatorname{conv}(x_k, s_k)}{\arg\min} f(x).$$

We use conv to denote the convex hull—in this last example, a line segment. Another variant, the *fully corrective* CGM, chooses

$$x_{k+1} = \underset{x \in \operatorname{conv}(x_k, s_1, \dots, s_k)}{\arg\min} f(x).$$

In the next section, we propose a natural choice for this step in the case of measures that use local search to speed up the convergence of the CGM.

One appealing aspect of the CGM is that it is very simple to compute a lower bound on the optimal value $f_\star$ as the algorithm runs. As $\hat{f}_k$ lower bounds $f$, we have

$$f(s) \geq \hat{f}_k = f(x_k) + f'(s - x_k; x_k) = \hat{f}_k(s)$$

for any $s \in \mathcal{C}$. Minimizing both sides over $s$ gives us the elementary bound

$$f_\star \geq \hat{f}_k(s_k).$$

The right-hand side of this inequality is readily computed after step 2. One can prove that the bound on suboptimality derived from this inequality decreases to zero [29], which makes it a very useful termination condition.

### 3.1. CGM for sparse inverse problems.

In this section we apply the classical CGM to the sparse inverse problem (1.4). We give two versions—first a direct translation of the fully corrective variant and then our improved algorithm that leverages local search on $\Theta$. To make it clear that we operate over the space of measures on $\Theta$, we change notation and denote the iterate by $\mu_k$ instead of $x_k$. The most obvious challenge is that we cannot easily represent a general measure on a computer. However, we will see that the steps of CGM can, in fact, be carried out on a computer in this context. In fact, each iterate is a sparse measure $\mu_k$ supported on $N_k$ points:

$$\mu_k = \sum_{i=1}^{N_k} w_i^{(k)} \delta_{\theta_i^{(k)}}.$$

As such, we will represent $\mu_k$ by the pair of vectors: $w_k \in \mathbb{R}^{N_k}$ and $\vec{\theta}_k \in \Theta^{N_k}$. Indeed, the algorithms we present can be thought of as operating on this representation directly.

Before we describe the algorithm in detail, we first explain how to linearize the objective function and minimize the linearization. In the space of measures, linearization is most easily understood in terms of the (one-sided) directional derivative.

In our formulation (1.4), $f(\mu) = \ell(\Phi\mu - y)$. If we define the *residual* as $r_k = \Phi\mu_k - y$, we can compute the directional derivative of our particular choice of $f$ at $\mu_k$ in the direction of the measure $s$ as

$$(3.2)\quad f'(s;\mu_k) = \lim_{t\swarrow 0} \frac{\ell(\Phi(\mu_k + ts) - y) - \ell(\Phi(\mu_k) - y)}{t}$$
$$= \lim_{t\swarrow 0} \frac{\ell(r_k + t\Phi s) - \ell(r_k)}{t} = \ell'(\Phi s; r_k) = \langle \nabla\ell(r_k), \Phi s\rangle .$$

Here, the inner product on the right-hand side of the equation is the standard inner product in $\mathbb{R}^d$.

The second step of the CGM minimizes the linearized objective over the constraint set. In other words, we minimize $\langle\nabla\ell(r_k), \Phi s\rangle$ over a candidate measure $s$ with total variation bounded by $\tau$. Interchanging the integral (in $\Phi$) with the inner product, and defining $F(\theta) := \langle\nabla\ell(r_k), \psi(\theta)\rangle$, we need to solve the optimization problem

$$(3.3)\qquad \underset{|s|(\Theta)\leq\tau}{\text{minimize}} \int F(\theta)ds(\theta).$$

An optimal solution of (3.3) is the point mass $-\tau\text{sgn}(F(\theta_\star))\delta_{\theta_\star}$, where $\theta_\star \in \arg\max$ $|F(\theta)|$. This is clear, as $\int F(\theta)ds(\theta)$ is bounded by $-\sup_\theta |F(\theta)|\|s\|_1$. This means that at each step of the CGM we need only add a single point to the support of our approximate solution $\mu_k$.

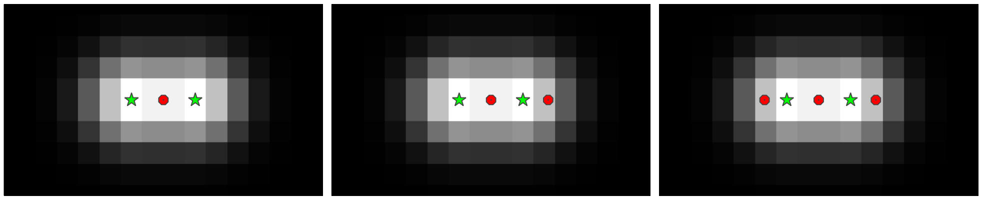
We now describe the fully corrective variant of the CGM for sparse inverse problems (Algorithm 2). The state of the algorithm at iteration $k$ is an atomic measure $\mu_k$ supported on a finite set $\vec{\theta}_k$ with weights $w_k$. The algorithm alternates between selecting a source to add to the support, and tuning the weights to lower the current cost. Step 4 is a finite-dimensional convex optimization problem that we can solve with an off-the-shelf algorithm. The solution to the finite-dimensional optimization problem may set some of $w_{k+1}$ to zero, in which case the prune subroutine removes the corresponding entries from $w_{k+1}$ and $\vec{\theta}_{k+1}$.

**Algorithm 2** Conditional gradient method for measures (CGM-M).

```
For $k = 1 : k_{\max}$
    1. Compute gradient of loss:    $g_k \leftarrow \nabla\ell(\Phi\mu_k - y)$.
    2. Compute next source:         $\theta_k \in \arg\max_{\theta\in\Theta} |\langle g_k, \psi(\theta)\rangle|$.
    3. Update support:              $\vec{\theta}_{k+1} \leftarrow [\vec{\theta}_k, \theta_k]$.
    4. Compute weights:             $w_{k+1} \leftarrow \arg\min_{\|w_{k+1}\|_1\leq\tau} \ell\left(\Phi\mu_{k+1} - y\right)$.
    5. Prune support:               $(w_{k+1}, \vec{\theta}_{k+1}) \leftarrow \text{prune}(w_{k+1}, \vec{\theta}_{k+1})$.
```

We stress here that the objective in step 2 is *nonlinear* (and nonconvex) in the parameter $\theta$, but *linear* when considered as a functional of the measure $s_k$.

While we can simply run for a fixed number of iterations, we may stop early using the standard CGM bound. With a tolerance parameter $\epsilon > 0$, we terminate



(a) (b) (c)

Fig. 1. *The three plots above show the first three iterates of the fully corrective CGM in a simulated superresolution imaging problem with two point sources of light. The locations of the true point sources are indicated by green stars, and the greyscale background shows the pixelated image. The elements of $S_k$ for $k = 1, 2, 3$ are displayed by red dots.*

when the conditional gradient bound assures us that we are at most $\epsilon$-suboptimal. In particular, we terminate when

$$\tau|\langle\psi(\theta_k), g_k\rangle| + \langle\Phi\mu_k, g_k\rangle < \epsilon. \tag{3.4}$$

Unfortunately, CGM-M does not perform well in practice. Not only does it converge very slowly, but the solution it finds is often supported on an undesirably large set. As illustrated in Figure 1, the performance of CGM-M is limited by the fact that it can only change the support of the measure by adding and removing points; it cannot smoothly move $S_k$ within $\Theta$. Figure 1 shows CGM-M applied to an image of two closely separated sources. The first source $\theta_1$ is placed in a central position overlapping both true sources. In subsequent iterations, sources are placed too far to the right and left, away from the true sources. To move the support of the candidate measure requires CGM-M to repeatedly add and remove sources; it is clear that the ability to move the support smoothly within the parameter space would resolve this issue immediately.

In practice, we can speed up convergence and find significantly sparser solutions by allowing the support to move continuously within $\Theta$. The following algorithm, which we call the ADCG, exploits the differentiability of $\psi$ to locally improve the support at each iteration.

**Algorithm 3** Alternating descent conditional gradient method (ADCG).

```
For $k = 1 : k_{\max}$
    1. Compute gradient of loss:      $g_k \leftarrow \nabla\ell(\Phi\mu_k - y)$.
    2. Compute next source:           $\theta_k \in \arg\max_{\theta\in\Theta} |\langle\psi(\theta), g_k\rangle|$.
    3. Update support:                $\vec{\theta}_{k+1} \leftarrow [\vec{\theta}_k, \theta_k]$.
    4. Coordinate descent on nonconvex objective:
       Repeat:
         (a) Compute weights:             $w_{k+1} \leftarrow \arg\min_{\|w_{k+1}\|_1\le\tau} \ell\left(\Phi\mu_{k+1} - y\right)$.
         (b) Prune support:               $(w_{k+1}, \vec{\theta}_{k+1}) \leftarrow \text{prune}(w_{k+1}, \vec{\theta}_{k+1})$.
         (c) Locally improve support:     $\vec{\theta}_{k+1} \leftarrow \textbf{local\_descent}(w_{k+1}, \vec{\theta}_{k+1})$.
```

Here **local_descent** is a subroutine that takes a measure $\mu$ with atomic representation $w, \vec{\theta}$ and attempts to use gradient information to reduce the function

$$(\theta_1, \ldots, \theta_m) \mapsto \ell\left(\sum_{i=1}^{m} w_i\psi(\theta_i) - y\right),$$

holding the weights fixed.

When the number of sources is held fixed, the optimization problem

$$\begin{array}{ll} \text{minimize} & \ell\left(\sum_{i=1}^{m} w_i \psi(\theta_i) - y\right) \\ \text{subject to} & \vec{\theta} \in \Theta^m, \\ & \|w\|_1 \leq \tau, \end{array} \tag{3.5}$$

is nonconvex. Step 4 is then block coordinate descent over $w$ and $\vec{\theta}$. The algorithm as a whole can be interpreted as alternating between performing descent on the convex (but infinite-dimensional) problem (1.4) in step 2 and descent over the finite-dimensional (but nonconvex) problem (3.5) in step 4. The bound (3.4) remains valid and can be used as a termination condition. Note that while we use block coordinate descent for simplicity, other algorithms that perform descent over $w$ and $\vec{\theta}$ simultaneously would also work well.

As we have previously discussed, this nonconvex local search does not change the convergence guarantees of the CGM whatsoever. We will show in Appendix B that this is an immediate consequence of the existing theory on the CGM. However, as we will show in section 5, the inclusion of local search dramatically improves the performance of the CGM.

**3.2. Interface and implementation.** Roughly speaking, running ADCG on a concrete instance of (1.4) requires subroutines for two operations. We need algorithms to compute

(a) $\psi(\theta)$ and $\frac{d}{d\theta}\psi(\theta)$ for $\theta \in \Theta$;

(b) $\arg\max_{\theta \in \Theta} |\langle \psi(\theta), v\rangle|$ for arbitrary vectors $v \in \mathbb{R}^d$.

Computing (a) is usually straightforward in applications with differentiable measurement models. Computing (b) is not easy in general. However, there are many applications of interest where (b) is tractable. For example, if the parameter space $\Theta$ is low dimensional, then the ability to compute (a) is sufficient to approximately compute (b): we can simply grid the parameter space and begin local search using the gradient of the function $\theta \mapsto \langle \psi(\theta), v\rangle$. Note that because of the local improvement step, ADCG works well even without exact minimization of (b). We prove this fact about inexact minimization in Appendix B.

If the parameter space is high dimensional, however, the feasibility of computing (b) will depend on the specific application. One example of particular interest that has been studied in the context of the CGM is matrix completion [30, 43, 25, 56]. In this case, the (b) step reduces to computing the leading singular vectors of a sparse matrix. We will show that adding local improvement to the CGM accelerates its convergence on matrix completion in the experiments.

We also note that in the special case of linear system identification, $\Theta$ is 6 dimensional, which is just large enough such that gridding is not feasible. In this case, we show that we can reduce the 6-dimensional optimization problem to a 2-dimensional problem and then again resort to gridding. We expect that in many cases of interest, such specialized solvers can be applied to solve the selection problem (b).

## 4. Related work.

There has recently been a renewed interest in the CGM as a general purpose solver for constrained inverse problems [29, 25]. These methods are simpler to implement than the projected or proximal gradient methods which require solving a quadratic rather than linear optimization over the constraint set.

The idea of augmenting the classical CGM with improvement steps is not unique to our work. Indeed, it is well known that any modification of the iterate that decreases the objective function will not hurt theoretical convergence rates [29]. Moreover, Rao, Shah, and Wright [43] have proposed a version of the CGM, called CoGENT, for atomic-norm problems that takes advantage of many common structures that arise in inverse problems. The reduction described in our theoretical analysis makes it clear that our algorithm can be seen as an instance of CoGENT specialized to the case of measures and differentiable measurement models.

The most similar proposals to ADCG come from the special case of matrix completion or nuclear-norm regularized problems. Several papers [56, 36, 25, 30] have proposed algorithms based on combinations of rank-one updates and local nonconvex optimization inspired by the well-known heuristic of [9]. While our proposal is significantly more general, ADCG essentially recovers these algorithms in the special case of nuclear-norm problems.

We note that in the context of inverse problems, there are a variety of algorithms proposed to solve the general infinite-dimensional problem (1.4). Tang, Bhaskar, and Recht [52] prove that this problem can be approximately solved by gridding the parameter space and solving the resulting finite dimensional problem. However, these gridding approaches are not tractable for problems with parameter spaces of even relatively modest dimension. Moreover, even when gridding is tractable, the solutions obtained are often supported on very large sets and heuristic postprocessing is required to achieve reasonable performance in practice [52]. In spite of these limitations, gridding is the state of the art in many application areas including computational neuroscience [19], superresolution fluorescence microscopy [57], radar [7, 26], remote sensing [21], compressive sensing [6, 38, 16], and polynomial interpolation [44].

There have also been a handful of papers that attempt to tackle the infinite-dimensional problem without gridding. For the special case where $\ell(\cdot) = \|\cdot\|_2^2$, Bredies and Pikkarainen [8] propose an algorithm to solve the Tikhonov-regularized version of problem (1.4) that is very similar to Algorithm 3. They propose performing a conditional gradient step to update the support of the measure, followed by soft thresholding to update the weights. Finally, with the weights of the measure fixed they perform discretized gradient flow over the locations of the point masses. However, they do not solve the finite-dimensional convex problem at every iteration, which means there is no guarantee that their algorithm has bounded memory requirements. Much more seriously, for the same reason, they are limited to one pass of gradient descent in the nonconvex phase of the algorithm. In section 5 we show that this limitation has serious performance implications in practice.

## 5. Numerical results.

In this section we apply ADCG to three of the examples in section 2: superresolution fluorescence microscopy, matrix completion, and system identification. We have made a simple implementation of ADCG publicly available on github: https://github.com/nboyd/SparseInverseProblems.jl. This allows the interested reader to follow along with these examples, and, hopefully, to apply ADCG to other instances of (1.4).

For each example we briefly describe how we implement the required subroutines for ADCG, though again the interested reader may want to consult our code for the full picture. We then describe how ADCG compares to prior art. Finally, we show how ADCG improves on the standard fully corrective CGM-M and a variant of the gradient flow (GF) algorithm proposed in [8]. While the GF algorithm proposed in [8] does not solve the finite-dimensional convex problem at each step, our version of GF does. We feel that this is a fair comparison: intuitively, fully solving the convex problem can only improve the performance of the GF algorithm. All three experiments require a subroutine to solve the finite-dimensional convex optimization problem over the weights. For this we use a simple implementation of a primal-dual interior point method, which we include in our code package.

For each experiment we select the parameter $\tau$ by inspection. For matrix completion and linear system identification this means using a validation set. For single molecule imaging each image requires a different value of $\tau$. For this problem, we run ADCG with a large value of $\tau$ and stop when the decrease in the objective function gained by the addition of a source falls below a threshold. This heuristic can be viewed as post hoc selection of $\tau$ and the stopping tolerance $\epsilon$, or as a stagewise algorithm [54].

The experiments are run on a standard c4.8xlarge EC2 instance. Our naive implementations are meant to demonstrate that ADCG is easy to implement in practice and finds high-quality solutions to (1.4). For this reason we do not include detailed timing information.

**5.1. Superresolution fluorescence microscopy.** We analyze data from the single molecule localization microscopy (SMLM) challenge [48, 23]. Fluorescence microscopy is an imaging technique used in the biological sciences to study subcellular structures in vivo. The task is to recover the 2-dimensional positions of a collection of fluorescent proteins from images taken through an optical microscope.

Here we compare the performance of ADCG to the gridding approach of Tang, Bhaskar, and Recht [52], two algorithms from the microscopy community (quickPALM and center of Gaussians), and also CGM and the GF algorithm proposed by [8]. The gridding approach approximately solves the continuous optimization problem (1.4) by discretizing the space $\Theta$ into a finite grid of candidate point source locations and running an $\ell_1$-regularized regression. In practice there is typically a small cluster of nonzero weights in the neighborhood of each true point source. With a fine grid, each of these clusters contains many nonzero weights, yielding many false positives.

To remove these false positives, Tang, Bhaskar, and Recht [52] propose a heuristic postprocessing step that involves taking the center of mass of each cluster. This postprocessing step is hard to understand theoretically, and does not perform well with a high density of fluorophores.

**5.1.1. Implementation details.** For this application, the minimization required in step 2 of ADCG is not difficult: the parameter space is 2-dimensional. Coarse gridding followed by a local optimization method works well in theory and practice.

For **local_descent** we use a standard constrained gradient method provided by the NLopt library [32].

**5.1.2. Evaluation.** We measure localization accuracy by computing the $F_1$ score, the harmonic mean of precision, and recall, at varying radii. Computing the precision and recall involves first matching estimated point sources to true point sources—a difficult task. Fortunately, the SMLM challenge website [23] provides a stand-alone application that we use to compute the $F_1$ score.

We use a dataset of 12000 images that overlay to form simulated microtubules (see Figure 2) available online at the SMLM challenge website [23]. There are 81049 point sources in total, roughly evenly distributed across the images. Figure 2(a) shows a typical image. Each image covers an area 6400 nm across, meaning each pixel is roughly 100 nm by 100 nm.

Figure 3 compares the performance of ADCG, gridding, quickPALM, and center of Gaussians (CoG) on this dataset. We match the performance of the gridding al-

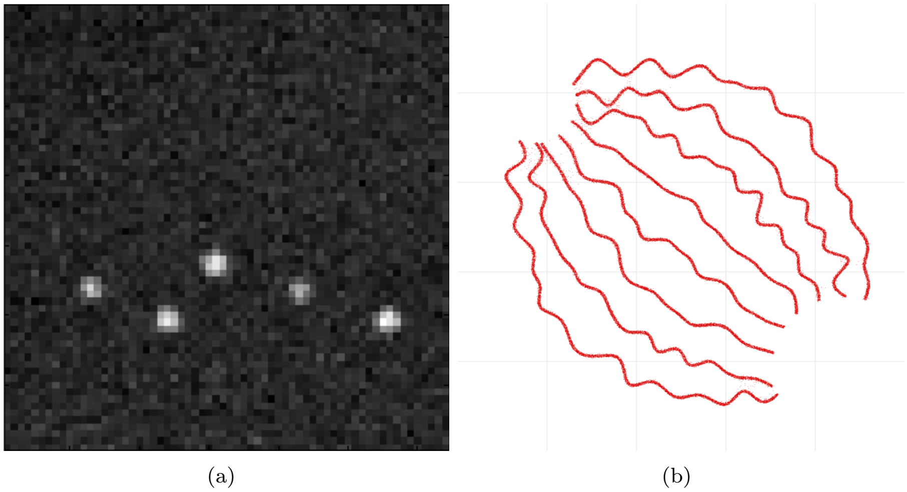


Fig. 2. *The long sequence dataset contains* 12000 *images similar to* (a). *The recovered locations for all the images are displayed in* (b).

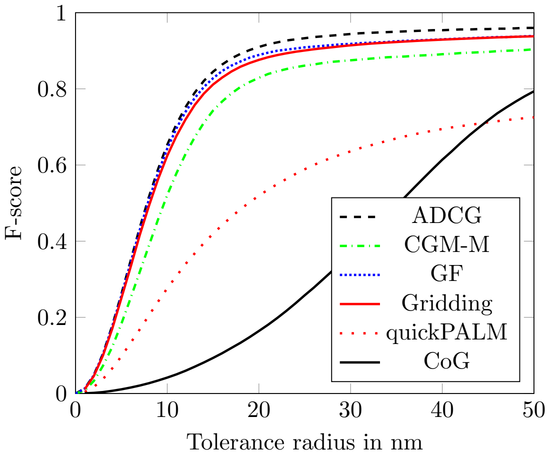


Fig. 3. *Performance on bundled tubes: long sequence. F-scores at various radii for* 6 *algorithms. For reference, each image is* 6400 *nm across, meaning each pixel has a width of* 100 *nm. ADCG outperforms all competing methods on this dataset.*

gorithm from [52], and significantly beat both quickPALM and CoG. Our algorithm analyzes all images in well under an hour—significantly faster than the gridding approach of [52]. Note that the gridding algorithm of [52] does not work without a postprocessing step.

**5.2. Matrix completion.** As described in section 2, matrix completion is the task of estimating an approximately low-rank matrix from some of its entries. We test our proposed algorithm on the Netflix Prize dataset, a standard benchmark for matrix completion algorithms.

**5.2.1. Implementation details.** Although the parameter space for this example is high dimensional we can still compute the steepest descent step over the space

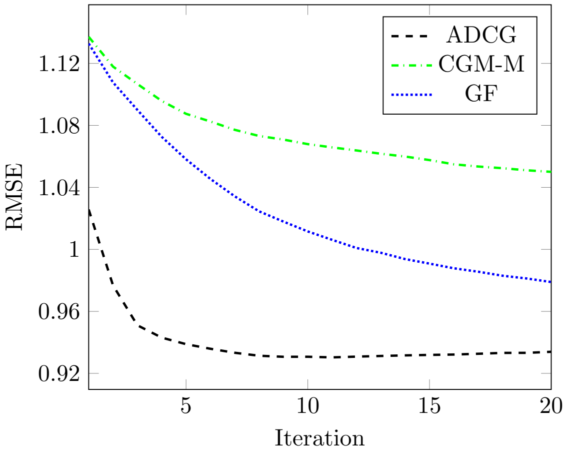


FIG. 4. *RMSE on Netflix Challenge dataset. ADCG significantly outperforms CGM-M.*

of measures. We need to minimize the following over $a, b$ with $\|a\|_2 = \|b\|_2 = 1$:

$$\langle \psi(a,b), \nu \rangle = \langle M(ab^T), \nu \rangle = \langle ab^T, M^*(\nu) \rangle = a^T M^*(\nu) b.$$

In other words, we need to find the unit norm, rank-one matrix with highest inner product with the matrix $M^*\nu$. The solution to this problem is given by the top singular vectors of $M^*\nu$. Computing the top singular vectors using a Lanczos method is relatively easy as the matrix $M^*\nu$ is extremely sparse.

Our implementation of **local_descent** takes a single step of gradient descent (on the sphere) with line search.

**5.2.2. Evaluation.** Our algorithm matches the state of the art for nuclear-norm-based approaches on the Netflix Prize dataset. Briefly, the task here is to predict the ratings 480189 Netflix users give to a subset of 17770 movies. One approach has been to phrase this as a matrix completion problem. That is, to try to complete the 480189 by 17770 matrix of ratings from the observed entires. Following [45] we subtract the mean training score from all movies and truncate the predictions of our model to lie between 1 and 5.

Figure 4 shows root-mean-square error (RMSE) of our algorithm and other variants of the CGM on the Netflix probe set. Again, ADCG outperforms all other CGM variants. Our algorithm takes over 7 hours to achieve the best RMSE—this could be improved with a more sophisticated implementation, or parallelization.

**5.2.3. Comparison to prior approaches.** Many researchers have proposed solving matrix completion problems or general semidefinite programs using CGM-like algorithms; see [56, 36, 25, 30]. While ADCG applied to the matrix completion problem is distinct (to the best of our knowledge) from existing algorithms, it combines existing ideas. For instance, the idea of using the conditional gradient algorithm to solve the constrained formulation is very well known [30]. The idea of using local search on a low-rank factorization goes back at least to [9], and is used in many recent algorithms [56, 36].

In terms of performance, our implementation is relatively slow but gives very good validation error.

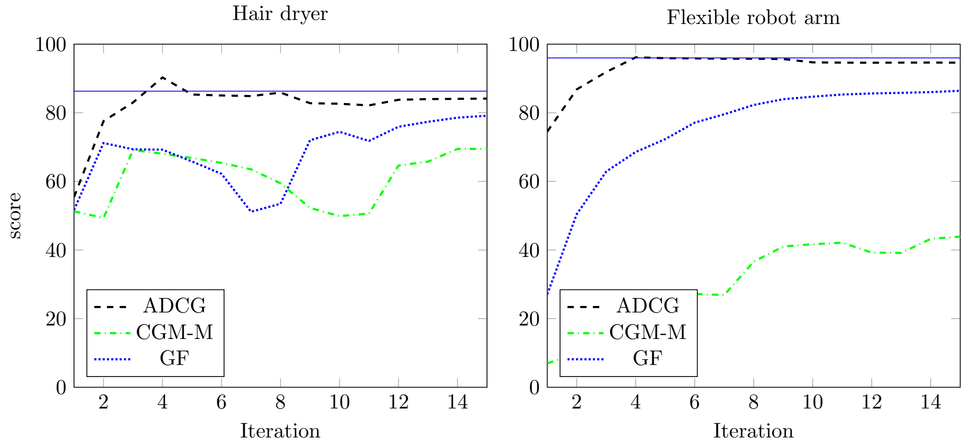


FIG. 5. *Performance on DaISy datasets. ADCG outperforms other CGM variants and matches the nuclear-norm-based technique of* [49].

**5.3. System identification.** In this section we apply our algorithms to identifying two single-input single-output systems from the DaISy collection [13]: the flexible robot arm dataset (ID 96.009) and the hairdryer dataset (ID 96.006).

**5.3.1. Implementation details.** While the parameter space is 6-dimensional, which effectively precludes gridding, we can efficiently solve the minimization problem in step (b) of the ADCG. To do this, we grid only over $r$ and $\alpha$: the output is linear in the remaining parameters ($B$ and $x_0$) allowing us to analytically solve for the optimal $B$ and $x_0$ as a function of $r$ and $\alpha$.

For **local_descent** we again use a standard box-constrained gradient method provided by the NLopt library [32].

**5.3.2. Evaluation.** Both datasets were generated by driving the system with a specific input and recording the output. The total number of samples is 1000 in both cases. Following [49] we identify the system using the first 300 time points and we evaluate performance by running the identified system forward for the remaining time points and compare our predictions to the ground truth.

We evaluate our predictions $y_{\text{pred}}$ using the score defined in [24]. The score is given by

$$\text{score} = 100\left(1 - \frac{\|y_{\text{pred}} - y\|_2}{\|y_{\text{mean}} - y\|_2}\right), \tag{5.1}$$

where $y_{\text{mean}}$ is the mean of the test set $y$.

Figure 5 shows the score versus the number of sources as we run our algorithm. For reference we display with horizontal lines the results of [24]. ADCG matches the performance of [24] and exceeds that of all other CGM variants. Our simple implementation takes about an hour, which compares very poorly with the spectral methods in [24] which complete in under a minute.

**6. Conclusions and future work.** As demonstrated in the numerical experiments of section 5, ADCG achieves state-of-the-art performance in superresolution fluorescence microscopy, matrix completion, and system identification, without the need for heuristic postprocessing steps. The addition of the nonconvex local search step **local_descent** significantly improves performance relative to the standard conditional gradient algorithm in all of the applications investigated. In some sense, we can understand ADCG as a method to rigorously control local search. One could just start with a model expansion (1.1) and perform nonconvex local search. However, this fares far worse than ADCG in practice and has no theoretical guarantees. The ADCG framework provides a clean way to generate a globally convergent algorithm that is practically efficient. Understanding this coupling between local search heuristics and convex optimization leads our brief discussion of future work.

*Tighten convergence analysis for ADCG.* The CGM is a robust technique, and adding our auxiliary local search step does not change its convergence rate. However, in practice, the difference between the ordinary CGM, the fully corrective variants, and ADCG are striking. In many of our experiments, ADCG outperforms the other variants by appreciable margins. Yet, all of these algorithms share the same upper bound on their convergence rate. A very interesting direction of future work would be to investigate if the bounds for ADCG can be tightened at all to be more predictive of practical performance. There may be connections between our algorithm and other alternating minimization techniques popular in matrix completion [33, 31], sparse coding [1, 2], and phase retrieval [39], and perhaps the techniques from this area could be applied to our setting of sparse inverse problems.

*Connections to clustering algorithms.* Another possible connection that could be worth exploring is the connection between the CGM and clustering algorithms like k-means. Theoretical bounds have been devised for initialization schemes for clustering algorithms that resemble the first step of CGM [3, 40]. In these methods, k-means is initialized by randomly seeking the points that are farthest from the current centers. This is akin to the first step of CGM which seeks the model parameters that best describe the residual error. Once a good seeding is acquired, the standard Lloyd iteration for k-means can be shown to converge to the global optimal solution [40]. It is possible that these analyses could be generalized to analyze our version of CGM or inspire new variants of the CGM.

*Connections to cutting plane methods and semi-infinite programs.* The standard Lagrangian dual of (1.4) is a semi-infinite program (SIP), namely, an optimization problem with a finite-dimensional decision variable but an infinite collection of constraints [27, 50]. One of the most popular algorithmic techniques for SIP is the cutting plane method, and these methods qualitatively act very much like the CGM. Exploring this connection in detail could generate variants of cutting plane methods suited for continuous constraint spaces. Such algorithms could be valuable tools for solving SIPs that arise in contexts disjoint from sparse inverse problems.

*Other applications.* We believe that our techniques are broadly applicable to other sparse inverse problems, and hope that future work will explore the usefulness of ADCG in areas unexplored in this paper. To facilitate the application of ADCG to more problems, such as those described in section 2, we have made our code publicly available on GitHub. As described in section 3, implementing ADCG for a new application essentially requires only two user-specified subroutines: one routine that evaluates the observation model and its derivatives at a specified set of weights and model parameters, and one that approximately solves the linear minimization in step 2 of ADCG. We aim to investigate several additional applications in the near future to test the breadth of the efficacy of ADCG.

## Appendix A. Relationship to atomic-norm problems.

Problems similar to (1.4) have been widely studied through the lens of atomic norms [12]. In this section we review the definition of an atomic norm and examine the intimate connection between (1.4) and a particular atomic-norm problem. We discuss the advantages and disadvantages of each formulation. Finally, we note that the linear invariance of the CGM links not only the optimization problems, but also the iterates of the CGM applied to each problem.

### A.1. An analogous atomic-norm problem.

The atomic norm $\|\cdot\|_{\mathcal{A}}$ corresponding to a suitable collection of atoms $\mathcal{A} \subset \mathbb{R}^d$ is the Minkowski functional of $\operatorname{conv}(\mathcal{A})$, defined by

$$\|x\|_{\mathcal{A}} = \inf\{\tau \geq 0 : x \in \tau \operatorname{conv}(\mathcal{A})\}.$$

Here $\operatorname{conv}(\mathcal{A})$ refers to the convex hull of $\mathcal{A}$. The atomic norm, while not properly a norm without additional restrictions [12], is always a convex function. Much research has gone into the problem of estimating a vector $x \in \mathbb{R}^d$ that is believed to be a sum of a few elements of $\mathcal{A}$, using the atomic norm as a regularizer [12, 43].

It is tempting to approach (1.2) using the atomic norm generated by the set $\mathcal{A} = \{\pm\psi(\theta) : \theta \in \Theta\}$. Indeed, the noiseless observation vector $y$ is a sum of a few (weighted) elements of $\mathcal{A}$. The atomic norm analogue to (1.4) is given by

$$\begin{array}{ll} \text{minimize} & \ell\,(x - y) \\ \text{subject to} & x \in \tau \operatorname{conv}(\mathcal{A}). \end{array} \tag{A.1}$$

We'll show that the infinite-dimensional optimization problem (1.4) and the finite-dimensional atomic-norm problem (A.1) are equivalent (in the sense of optimal objective value) under the (linear) change of variables

$$\mu \mapsto \Phi\mu. \tag{A.2}$$

Lemma A.1 shows that the feasible set of (A.1) is exactly the image of the feasible set of (1.4) under the linear transformation $\Phi$, which implies that the optimal objective values of (1.4) and (A.1) are the same, and that the solutions are linked by $x_\star = \Phi\mu_\star$.

Lemma A.1. $\operatorname{conv}(\mathcal{A}) = \{\Phi\mu : \|\mu\|_1 \leq 1\}$.

*Proof.* It's clear that $\operatorname{conv}(\mathcal{A}) \subset \{\Phi\mu : \|\mu\|_1 \leq 1\}$: the convex hull is the smallest convex set containing $\{\pm\psi(\theta) : \theta \in \Theta\}$, which are the images of the measures $\pm\delta_\theta$ under $\Phi$.

The other direction is slightly more difficult. Suppose $y = \int \psi(\theta) d\mu(\theta)$ with $\|\mu\|_1 \leq 1$ and $y \notin \operatorname{conv}(\mathcal{A})$. Now we'd like to strongly separate $\{y\}$ from $\operatorname{conv}(\mathcal{A})$ using Corollary 1.4.2 from [46]. This requires $y$ to be outside of the closure of $\operatorname{conv}(\mathcal{A})$. Fortunately, $\operatorname{conv}(\mathcal{A})$ is already closed. To see this, note that $\mathcal{A}$ is compact as it is the union of two sets, each of which is compact as they are the image of the compact set $\Theta$ under the continuous function $\psi$. Finally, as $\mathcal{A}$ is a compact set in a finite-dimensional space, Caratheodory's theorem implies that its convex hull is also compact [46, Theorem 17.2].

Then by [46, Corollary 1.4.2], there exists a hyperplane separating $y$ and $\operatorname{conv}(\mathcal{A})$ strongly. Let $v$ be the normal vector to the separating hyperplane. Then we have $\langle v, y\rangle > \sup_\theta \pm\langle v, \psi(\theta)\rangle = \sup_\theta |\langle v, \psi(\theta)\rangle|$. This is a contradiction as $\langle v, y\rangle = \langle v, \int \psi(\theta) d\mu(\theta)\rangle = \int \langle v, \psi(\theta)\rangle d\mu(\theta) \leq \sup_\theta |\langle v, \psi(\theta)\rangle| \|\mu\|_1 \leq \sup_\theta |\langle v, \psi(\theta)\rangle|$. □

**A.2. Discussion.** Much of the literature on sparse inverse problems focuses on problem (A.1), as opposed to the infinite-dimensional problem (1.4). This focus is due to the fact that (A.1) has algorithmic and theoretical advantages over (1.4). First and foremost, (A.1) is finite dimensional, which means that standard convex optimization algorithms may apply. Additionally, the geometry of the atomic-norm ball, $\operatorname{conv}\{\pm\psi(\theta) : \theta \in \Theta\}$, gives clean geometric insight into when the convex heuristic will work [12].

With that said, we hold that the infinite-dimensional formulation we study has distinct practical and theoretical advantages over the atomic-norm problem (A.1), at least when one is interested in (1.3). A solution $x_\star$ to the atomic-norm problem is an estimate of the vector $\Phi\mu_{\text{true}}$, while what we actually seek is an estimate of the signal parameters $\{(w_1, \theta_1), \ldots, (w_k, \theta_k)\}$. In many applications, it is this atomic decomposition $\mu$ that is of interest, and *not* the optimal point $x_\star$ of (A.1). Reconstructing an optimal $\mu_\star$ for problem (1.4) from $x_\star$ can be highly nontrivial; for many measurement models this is as hard as solving (1.4). As such, an algorithm that returns $x_\star$ is useless in the context of solving the original signal estimation task. For example, when designing radiation therapy, the measure $\mu_\star$ encodes the optimal beam plan directly, while the vector $x_\star = \Phi\mu_\star$ is simply the pattern of radiation that the optimal plan produces. Likewise, in superresolution microscopy, $\mu_\star$ encodes the locations and intensities of the localized fluorophores, while $x_\star$ is the *blurry* image they produce. For this reason, an algorithm that simply returns the vector $x_\star$, without the underlying atomic decomposition, is not always useful in practice.

Additionally, the measure-theoretic framework exposes the underlying parameter space, which in many applications comes with meaningful and useful structure. Furthermore, in many applications the measurement operator $\psi$ is known, but there may be no way to compute $\|x\|_{\mathcal{A}}$—meaning that many standard convex optimization algorithms cannot be immediately applied to the atomic-norm problem. Finally, naïve-interpretation of the finite-dimensional optimization problem treats the parameter space as an unstructured set; keeping the structure of the parameter space in mind makes extensions such as ADCG that make local movements in parameter space natural and uniform across applications.

**A.3. Invariance of the CGM under linear transformations.** Readers familiar with the literature on conditional gradient variants for atomic-norm problems (e.g., [43]) will notice that the iterations of ADCG look very similar to the iterations of the CGM applied to the atomic-norm problem (A.1). This is a result of the affine invariance property of the CGM discussed in [29]. Essentially, this property means that the iterates of the standard CGM applied to (1.4) are, after transformation by $\Phi$, the same as the iterates of the standard CGM applied to (A.1). Due to the fact that the theoretical analysis of the CGM still applies to algorithms that decrease the objective value at least as much as the standard CGM [29, 43], this means that the convergence rate of CGM variants, like ADCG, applied to (1.4) can be bounded by the convergence rate of the standard CGM applied to (A.1). Note that while this equivalence means that ADCG can be interpreted as solving the atomic-norm problem, it does not mean that general algorithms for solving the atomic norm problem can be interpreted as solving (1.4).

**Appendix B. Theoretical guarantees.** In this section we present two simple theoretical results. The first guarantees that we can run our algorithm with bounded memory—though as noted in section 1, the bound is of limited use in practice, where our algorithm (typically) terminates in far fewer than $d + 1$ iterations. The second result guarantees that the algorithm converges to an optimal point and bounds the worst-case rate of convergence. Again, though, this rate of convergence is much slower than the rate observed in practice. We emphasize that the main contribution of this paper is an algorithm that is effective in practice, and both of these guarantees are immediate consequences of existing theory.

**B.1. Bounded memory.** As the CGM for measures adds one point to the support of the iterate per iteration, we know that the cardinality of the support of $\mu_k$ is bounded by $k$. For large $k$, then, $\mu_k$ could have large support. The following theorem guarantees that we can run our algorithm with bounded memory and, in fact, we need only store at most $d+1$ points, where $d$ is the dimension of the measurements.

THEOREM B.1. *ADCG may be implemented to generate iterates with cardinality of support uniformly bounded by* $d+1$.

*Proof.* Lemma (B.2) allows us to conclude that the fully corrective step ensures that the support of the measure remains bounded by $d+1$ for all iterations. □

LEMMA B.2. *The finite-dimensional problem*

$$\underset{\|w\|_1 \le \tau}{\text{minimize}} \ \ell\left(\sum_i w_i \psi(\theta_i) - y\right) \tag{B.1}$$

*has an optimal solution* $w_\star$ *with at most* $d+1$ *nonzeros.*

*Proof.* Let $u_\star$ be any optimal solution to (B.1). As $u_\star$ is feasible, we have that

$$v = \sum_i u_{\star i}\psi(\theta_i) \in \tau\text{conv}(\{\pm\psi(\theta_i) : i = 1, \dots, m\}).$$

In other words, $\frac{v}{\tau}$ lies in the convex hull of a set in $\mathbb{R}^d$. Caratheodory's theorem immediately tells us that $\frac{v}{\tau}$ can be represented as a convex combination of at most $d+1$ points from $\{\pm\psi(\theta_i) : i = 1, \dots, m\}$. That is, there exists a $w_\star$ with at most $d+1$ nonzeros such that

$$\sum_{i=1}^m w_{\star i}\psi(\theta_i) = v.$$

This implies that $w_\star$ is also optimal for (B.1). □

Note that in order to find $w_\star$, we need to either use a simplex-type algorithm to solve (B.1) or explore the optimal set using the random ray-shooting procedure as described in [51].

**B.2. Convergence analysis.** We now analyze the worst-case convergence rate for ADCG applied to (1.4). Note that the discussion in the last section on atomic norms (Appendix A) implies that we can bound the convergence of ADCG by the convergence of the standard CGM applied to (A.1). With that said, standard proofs of the convergence of the CGM (known since the 1960s) [18, 15, 29] apply to any optimization problem of the following form in any Banach space:

$$\underset{x \in \mathcal{S}}{\text{minimize}}\, f(x). \tag{B.2}$$

Here $\mathcal{S}$ is a bounded convex set, and $f$ is a convex function. The convergence result depends on a curvature parameter of the function $f$ over the set $\mathcal{S}$, $C_f$. $C_f$ is a constant such that the following inequality is satisfied for all $x, s \in \mathcal{S}$ and $\eta \in (0,1)$:

$$f(x + \eta(s-x)) \le f(x) + \eta f'(s-x; x) + \frac{C_f}{2}\eta^2.$$

Intuitively, $C_f$ controls the quality of the linear approximation to $f$ made in each iteration of the CGM. The standard convergence result is stated below.

THEOREM B.3. *Let $C_f$ be the curvature parameter of the convex function $f$ on the bounded, convex set $\mathcal{C}$. Let $x_1, x_2, \ldots$ be the iterates of the standard CGM applied to* (B.2). *Let $f_\star$ be the optimal value of* (B.2). *Then we have*

$$f(x_k) - f_\star \leq \frac{C_f}{k+2}.$$

In our setting we can simply take $\mathcal{S} = \{\mu : \|\mu\|_1 \leq \tau\}$ and $f(\mu) = \ell(\Phi\mu - y)$. The affine invariance of the CGM implies that the curvature of $f$ over $\mathcal{S}$ is equal to the curvature of the function $g(x) = \ell(x - y)$ over the set $\mathcal{A} = \operatorname{conv}(\{\pm\psi(\theta) : \theta \in \Theta)\})$. As noted in [29], Theorem B.3 applies to any algorithm that reduces the objective value at each iteration at least as much as the standard CGM. As ADCG falls into this category, it also converges at the specified rate.

The theorem applies even when the linear minimization step is performed approximately [29]. That is, we allow $\theta_k$ to be chosen such that

$$|\langle \psi(\theta_k), g_k\rangle| \leq \max_{\theta\in\Theta} |\langle \psi(\theta), g_k\rangle| + \frac{\zeta}{k+2} \tag{B.3}$$

for some $\zeta \geq 0$. When inequality (B.3) holds, we say that the linear minimization problem in iteration $k$ is solved to precision $\zeta$. When the linear minimization problem is solved to precision $\zeta$, the convergence rate above applies when multiplied by the factor $(1 + \zeta)$.

**Acknowledgments.** We would like to thank Elina Robeva and Stephen Boyd for many useful conversations about this work.

## REFERENCES

[1] A. Agarwal, A. Anandkumar, P. Jain, and P. Netrapalli, *Learning sparsely used overcomplete dictionaries via alternating minimization*, SIAM J. Optim., 26 (2016), pp. 2775–2799.

[2] S. Arora, R. Ge, T. Ma, and A. Moitra, *Simple, Efficient, and Neural Algorithms for Sparse Coding*, preprint, arXiv:1503.00778, 2015.

[3] D. Arthur and S. Vassilvitskii, *k-means++: The advantages of careful seeding*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2007, pp. 1027–1035.

[4] F. Bach, *Convex relaxations of structured matrix factorizations*, arXiv:1309.3117, 2013, pp. 1355–1362.

[5] F. Bach, S. Lacoste-Julien, and G. Obozinski, *On the equivalence between herding and conditional gradient algorithms*, in ICML, Omnipress, Madison, WI, 2012, pp. 1355–1362.

[6] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak., *Compressed channel sensing: A new approach to estimating sparse multipath channels*, Proc. IEEE, 98 (2010), pp. 1058–1076.

[7] R. Baraniuk and P. Steeghs., *Compressive radar imaging*, In IEEE Radar Conference, Waltham, MA, IEEE, Piscataway, NJ, 2007, pp. 128–133.

[8] K. Bredies and H. K. Pikkarainen, *Inverse problems in spaces of measures*, ESAIM Control Optim. Calc. Var., 19 (2013), pp. 190–218, https://doi.org/10.1051/cocv/2011205.

[9] S. Burer and R. D. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357, https://doi.org/10.1007/s10107-002-0352-8.

[10] E. Candes and C. Fernandez-Granda., *Towards a mathematical theory of super resolution*, Comm. Pure Appl. Math, 67 (2014), pp. 906–956.

[11] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Commun. ACM, 55 (2012), pp. 111–119.

[12] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, *The convex geometry of linear inverse problems*, Found. Comput. Math., 12 (2012), pp. 805–849.

[13] B. L. R. De Moor, ed., *DaISy*, http://homes.esat.kuleuven.be/~smc/daisy/.

[14] B. G. R. de Prony, *Essai experimental et analytique: Sur les lois de la dilatabilite de fluides elastique et sur celles de la force expansive de la vapeur de l'alkool, a differentes temperatures*, J. Éc. Polytech., 1 (1795), pp. 24–76.

[15] V. Demyanov and A. Rubinov, *Approximate methods in optimization problems*, in Modern Analytic and Computational Methods in Science and Mathematics, Elsevier, New York, 1970.

[16] M. Duarte and R. Baraniuk, *Spectral compressive sensing*, Appl. Comput. Harmon. Anal., 35 (2013), pp. 111–129.

[17] N. Dunford, J. T. Schwartz, W. G. Bade, and R. G. Bartle, *Linear Operators, Part* 1, Wiley-interscience New York, 1958.

[18] J. Dunn and S. Harshbarger, *Conditional gradient algorithms with open loop step size rules*, J. Math. Anal. Appl., 62 (1978), pp. 432–444.

[19] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, *Neural spike identifcation with continuous basis pursuit*, Computational and Systems Neuroscience (CoSyNe), Salt Lake City, Utah, 2011.

[20] D. Evanko, *Primer: Fluorescence imaging under the diffraction limit*, Nature Methods, 6 (2009), pp. 19–20.

[21] A. C. Fannjiang, T. Strohmer, and P. Yan, *Compressed remote sensing of sparse objects*, SIAM J. Imaging Sci., 3 (2010), pp. 595–618.

[22] G. B. Folland, *Real Analysis*, Wiley, New York, 1999.

[23] B. I. Group, *Single-Molecule Localization Microscopy Benchworking Software*, http://bigwww.epfl.ch/palm/ (2013).

[24] A. Hansson, Z. Liu, and L. Vandenberghe, *Subspace System Identification via Weighted Nuclear Norm Optimization*, preprint, arXiv:1207.0023, 2012.

[25] Z. Harchaoui, A. Juditsky, and A. Nemirovski, *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Math. Program., 152 (2014), pp. 75–112.

[26] M. Herman and T. Strohmer, *High-resolution radar via compressed sensing*, IEEE Trans. Signal Process., 57 (2009), pp. 2275–2284.

[27] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods, and applications*, SIAM Rev., 35 (1993), pp. 380–429.

[28] H. Hindi, *A tutorial on optimization methods for cancer radiation treatment planning*, in American Control Conference (ACC), IEEE, Piscataway, NJ, 2013, pp. 6804–6816, https://doi.org/10.1109/ACC.2013.6580908.

[29] M. Jaggi, *Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization*, in ICML, 2013, Curran, Red Hook, NY, pp. I-427–I-435.

[30] M. Jaggi and M. Sulovský, *A simple algorithm for nuclear norm regularized problems*, in Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, Madison, WI, 2010, pp. 471–478.

[31] P. Jain, P. Netrapalli, and S. Sanghavi, *Low-rank matrix completion using alternating minimization*, in Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, ACM, New York, 2013, pp. 665–674.

[32] S. G. Johnson, *NLopt*, http://ab-initio.mit.edu/nlopt (2011).

[33] R. H. Keshavan, *Efficient Algorithms for Collaborative Filtering*, Ph.D. thesis, Stanford University, Palo Alto, CA, 2012.

[34] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*, The MIT Press, Cambridge, MA, 2009.

[35] S. Lacoste-Julien, F. Lindsten, and F. Bach, *Sequential kernel herding: Frank-Wolfe optimization for particle filtering*, J. Mach. Learn. Res. W&CP, 38 (2015), pp. 544–552.

[36] S. Laue, *A hybrid algorithm for convex semidefinite optimization*, in Proceedings of the 29th International Conference on Machine Learning (ICML-12), J. Langford and J. Pineau, eds., New York, 2012, Omnipress, Madison, WI, 2012, pp. 177–184.

[37] H.-Y. Liu, E. Jonas, L. Tian, J. Zhong, B. Recht, and L. Waller, 3*d imaging in volumetric scattering media using phase-space measurements*, Opt. Express, 23 (2015), pp. 14461–14471, https://doi.org/10.1364/OE.23.014461.

[38] D. Malioutov, M. Çetin, and A. S. Willsky, *A sparse signal reconstruction perspective for source localization with sensor arrays*, IEEE Trans. Signal Process., 53 (2005), pp. 3010–3022.

[39] P. Netrapalli, P. Jain, and S. Sanghavi, *Phase retrieval using alternating minimization*, in Advances in Neural Information Processing Systems, 26 (2013), pp. 2796–2804.

[40] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, *The effectiveness of Lloyd-type methods for the k-means problem*, J. ACM, 59 (2012), 28.

[41] F. Pukelsheim, *Optimal Design of Experiments*, Classics Appl. Math. 50, SIAM, Philadelphia, 2006.

[42] K. G. Puschmann and F. Kneer, *On super-resolution in astronomical imaging*, Astron. Astrophys., 436 (2005), pp. 373–378.

[43] N. Rao, P. Shah, and S. Wright, *Forward-Backward Greedy Algorithms for Atomic Norm Regularization*, prprint, arXiv:1404.5692, 2014.

[44] H. Rauhut, *Random sampling of sparse trigonometric polynomials*, Appl. Comput. Harmon. Anal., 22 (2007), pp. 16–42.

[45] B. Recht and C. Ré, *Parallel stochastic gradient algorithms for large-scale matrix completion*, Math. Program. Comput., 5 (2013), pp. 201–226.

[46] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[47] M. Rust, M. Bates, and X. Zhuang, *Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)*, Nature Methods, 3 (2006), pp. 793–796.

[48] D. Sage, H. Kirshner, T. Pengo, N. Stuurman, J. Min, S. Manley, and M. Unser, *Quantitative evaluation of software packages for single-molecule localization microscopy*, Nature Methods, 12 (2015), pp. 717–724, http://dx.doi.org/10.1038/nmeth.3442.

[49] P. Shah, B. Bhaskar, G. Tang, and B. Recht, *Linear System Identification via Atomic Norm Regularization*, preprint, arXiv:1204.0590, 2012.

[50] A. Shapiro, *Semi-infinite programming, duality, discretization and optimality conditions*, Optimization, 58 (2009), pp. 133–161.

[51] J. Skaf and S. Boyd, *Techniques for exploring the suboptimal set*, Optim. Eng., 11 (2010), pp. 319–337.

[52] G. Tang, B. Bhaskar, and B. Recht, *Sparse recovery over continuous dictionaries: Just discretize*, in 2013 Asilomar Conference on Signals, Systems, and Computers, IEEE, Piscataway, NJ, 2013, pp. 1043–1047.

[53] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, *Compressed sensing off the grid*, IEEE Trans. Inform. Theory, 59 (2013), pp. 7465–7490.

[54] R. J. Tibshirani, *A general framework for fast stagewise algorithms*, J. Mach. Learn. Res., 16 (2015), pp. 2543–2588.

[55] E. van den Berg and M. P. Friedlander, *Sparse optimization with least-squares constraints*, SIAM J. Optim., 21 (2011), pp. 1201–1229.

[56] X. Zhang, Y. Yu, and D. Schuurmans, *Accelerated training for matrix-norm regularization: A boosting approach*, in Advances in Neural Information Processing Systems 25, 2012, pp. 2906–2914.

[57] L. Zhu, W. Zhang, D. Elnatanand., and B. Huang, *Faster storm using compressed sensing*, Nature Methods, 9 (2012), pp. 721–723.