

The Alternating Descent Conditional Gradient Method for Sparse Inverse Problems

Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht

July 7, 2015

Abstract

We propose a variant of the classical conditional gradient method (CGM) for sparse inverse problems with differentiable measurement models. Such models arise in many practical problems including superresolution, 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 measurement 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.

1 Introduction

A ubiquitous prior in modern statistical signal processing asserts that an observed signal is the noisy measurement 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 non-parametric 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 and positions of aircraft—the space of sparse 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 [6, 32, 46]. In certain special cases, the infinite-dimensional optimization problem over measures can be reduced to a problem of moment estimation, and spectral techniques or semidefinite programming can be employed [19, 35, 47, 9]. More recently, in light of much of the work on compressed sensing and its generalizations, another proposal operates on atomic norms over data [11], 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 post-processing [46]. 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.

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 the 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 the 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 known as the Frank-Wolfe algorithm. In §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 §5, where we bound the convergence rate of our algorithm and also guarantee that it can be run with bounded memory. Moreover, in §6 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 non-negative weight $w > 0$, and a parameter $\theta \in \Theta$. An element θ of the parameter space Θ may describe, for instance, the position, orientation, and polarization of a source. The weight may encode the intensity of a source, or the distance of a source from the measurement 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 measurement in \mathbb{R}^d .

The measurement model we use is completely determined by a function $\psi : \Theta \rightarrow \mathbb{R}^d$, which gives the d -dimensional measurement of a single, unit-weight source parameterized by a point in Θ . The measurement of a lone source is homogeneous of degree one in its weight; that is, a single source with parameter θ and weight $w > 0$ generates the measurement $w\psi(\theta) \in \mathbb{R}^d$. Finally, we assume that the measurement of a weighted collection of sources is additive. In other words, the (noise-free) measurement of a weighted collection of sources, $\{(w_i, \theta_i)\}_{i=1}^K$, is simply

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

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 *measurement*.

We can encode the signal parameters as an atomic measure μ on Θ , with mass w_i at point θ_i : $\mu = \sum_{i=1}^K w_i \delta_{\theta_i}$. As a consequence of the additivity and homogeneity in our measurement model, the total measurement of a collection of sources encoded in the measure μ is a linear function Φ of μ :

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

We call Φ 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 Θ .

We now introduce the sparse inverse problem as an optimization problem over measures. Our goal is to recover μ_{true} from a measurement

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

corrupted by a noise term, ν . Recovering the signal parameters without any prior information is, in most interesting problems, impossible; the operator Φ 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 can assume that μ_{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 measurement model. In §2 we give several examples that are not recovery problems.

We estimate the signal parameters encoded in μ_{true} by minimizing a convex loss ℓ of the residual between y and $\Phi\mu$:

$$\begin{aligned} & \text{minimize} && \ell(\Phi\mu - y) \\ & \text{subject to} && \mu \geq 0 \\ & && |\text{supp}(\mu)| \leq N, \end{aligned} \tag{1.2}$$

where the optimization is over the infinite-dimensional space of measures on Θ . For example, when ℓ is the negative log-likelihood of the noise term ν , problem (1.2) corresponds to maximum likelihood estimation of μ_{true} . Here N is a posited upper bound on the size of the support of the true measure μ_{true} , which we denote by $\text{supp}(\mu_{\text{true}})$. Although here and elsewhere in the paper we explicitly enforce the constraint that the measure be nonnegative, all of our discussion and algorithms can be easily extended to the unconstrained case.

While the objective function in (1.2) is convex, the constraint on the support of μ is nonconvex. A common heuristic in these situations is to replace the nonconvex constraint with a convex surrogate. The standard surrogate for a cardinality constraint on a nonnegative measure is a constraint on the total mass. This substitution results in the standard convex approximation to (1.2):

$$\begin{aligned} & \text{minimize} && \ell(\Phi\mu - y) \\ & \text{subject to} && \mu \geq 0 \\ & && \mu(\Theta) \leq \tau. \end{aligned} \tag{1.3}$$

Here $\tau > 0$ is a parameter that controls the total mass of μ and empirically controls the cardinality of solutions to (1.3). While problem (1.3) 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 (1.3) may have arbitrarily large support, though we prove in §5 that we can always find solutions supported on at most $d+1$ points. In practice, however, we are interested in approximate solutions of (1.3) supported on far fewer than $d+1$ points.

A celebrated example of (1.3) occurs when Θ is the finite set $\{1, \dots, k\}$ and $\ell(r) = \frac{1}{2}\|r\|_2^2$. In that case, a nonnegative measure over Θ can be represented as a vector v in \mathbb{R}_+^k and the forward operator Φ as a matrix in $\mathbb{R}^{d \times k}$. The total mass of the nonnegative measure v is then simply $\sum_i v_i = \|v\|_1$. In this case, (1.3) reduces to the nonnegative lasso.

In this paper, we propose an algorithm for the substantially different case where Θ has some differential structure. Our algorithm is based on a variant of the conditional gradient method that takes advantage of the differentiable nature of ψ , and is guaranteed to produce approximate solutions with bounded support.

1.2 Relationship to atomic norm problems

Problems similar to (1.3) have been studied through the lens of atomic norms [11]. The atomic norm $\|\cdot\|_{\mathcal{A}}$ corresponding to a suitable collection of atoms $\mathcal{A} \subset \mathbb{R}^d$ is defined as

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

The connection to (1.3) becomes clear if we take $\mathcal{A} = \{\psi(\theta) : \theta \in \Theta\} \cup \{0\}$. With this choice of atomic set, we have the equality

$$\|x\|_{\mathcal{A}} = \inf \left\{ \mu(\Theta) : x = \int \psi(\theta) d\mu(\theta), \mu \geq 0 \right\}.$$

This equality implies the equivalence (in the sense of optimal objective value) of the infinite-dimensional optimization problem (1.3) to the finite-dimensional atomic norm problem:

$$\begin{aligned} & \text{minimize} && \ell(x - y) \\ & \text{subject to} && \|x\|_{\mathcal{A}} \leq \tau. \end{aligned} \tag{1.4}$$

Much of the literature on sparse inverse problems focuses on problem (1.4), as opposed to the infinite-dimensional problem (1.3). This focus is due to the fact that (1.4) has algorithmic and theoretical advantages over (1.3). First and foremost, (1.4) is finite-dimensional, which means that standard convex optimization algorithms may apply. Additionally, the geometry of the atomic norm ball, $\text{conv}\{\psi(\theta) : \theta \in \Theta\}$, gives clean geometric insight into when the convex heuristic will work [11].

With that said, we hold that the infinite-dimensional formulation we study has distinct practical advantages over the atomic norm problem (1.4). In many applications, it is the atomic decomposition that is of interest, and *not* the optimal point x_* of (1.4); reconstructing the optimal μ_* for problem (1.3) from x_* can be highly nontrivial. For example, when designing radiation therapy, the measure μ_* encodes the optimal beam plan directly, while the vector $x_* = \Phi\mu_*$ is simply the pattern of radiation that the optimal plan produces. For this reason, an algorithm that simply returns the vector x_* , 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—and is oftentimes more intuitive for practitioners than the corresponding atomic norm. 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.

2 Example applications

Many practical problems can be formulated as instances of (1.3). 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 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 $\theta_1, \dots, \theta_K$ denote the locations of K point sources (in \mathbb{R}^2 or \mathbb{R}^3), and w_i 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 ψ is the pixelated point spread function of the imaging apparatus.

By solving a version of (1.3) 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 [37]. In biology this tool has revolutionized imaging of subcellular features [16, 41]. A variant of this framework allows imaging through scattering media [31]. In §6.1, 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, \dots, u_T and the observed sequence of outputs y_1, \dots, y_T [43, 19]. 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 Θ is given by tuples of the form (x_0, r, α, B) where x_0 and B both lie in the ℓ_∞ unit ball in \mathbb{R}^2 , r is in $[0, 1]$, and α is in $[0, \pi]$. The LTI system that each source describes has

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

The mapping ψ from the parameters (x_0, r, α, B) to the output of the corresponding LTI system on input u_1, \dots, 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 §6.1, 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 [10]. 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 $\Theta = \{(u, v) \in \mathbb{R}^n \times \mathbb{R}^m : \|u\|_2 = \|v\|_2 = 1\}$, $\psi((u, v)) = M(uv^T)$, and $\ell(\cdot) = \|\cdot\|^2$. In §6.1, 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 $f : \Theta \rightarrow \mathbb{R}^d$ is a known differentiable feature function and ϵ_i are independent noise terms. We want to choose $\theta_1, \dots, \theta_k$ 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 [36].

In general, this task is intractable. However, if we assume ϵ_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_m is normal.

One notion of how much information y_1, \dots, y_m 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 sparse measure with bounded total mass, giving us an instance of (1.3).

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 i.i.d. samples. For background see [29]. 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 π . If we assume π 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 ℓ is the (negative) log-likelihood of the data: i.e., $y = 0$, $\ell(p) = -\sum_i \log p_i$. The obvious constraint is $\int d\pi(\theta) \leq 1$.

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, ψ is differentiable. A common choice of ℓ for this application is simply the squared loss. Note that in this application there is no need to constraint the weights to be positive.

Neural spike identification. 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 K 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 $\psi : \mathbb{R} \times \{1, \dots, K\} \rightarrow \mathbb{R}^d$ is a vector of voltage samples. 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 $\theta_1, \dots, \theta_K$ and weights w_1, \dots, w_K from the voltage signal v . For background see [15].

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 parameters θ 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 $\theta_1, \dots, \theta_K$ and intensities w_1, \dots, w_K add linearly, and the objective function is convex. For background see [23].

3 Conditional gradient method

In this section we present our main algorithmic development. We begin with a review of the classical conditional gradient method (CGM) for finite-dimensional convex programs. We then translate the classical CGM for the sparse inverse problem (1.3). In particular, we augment this algorithm with an aggressive 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 closed, bounded, and convex set and f is a differentiable convex function.

CGM proceeds by iteratively solving linearized versions of (3.1). At iteration k , we form a linear approximation to the function f at the current point x_k . We then minimize the linearization over the feasible set to get a potential solution s_k . This step can be interpreted as a restricted steepest descent: the linear functional represented by the gradient is simply the directional derivative. 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: $\hat{f}_k(s) \leftarrow f(x_k) + \langle \nabla f(x_k), s - x_k \rangle$.
 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 the tentative update in step 4, 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:

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

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

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

In the next section, we propose a natural choice for this step in the case of measures that uses 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_* as the algorithm runs. By convexity of f , we have

$$f(s) \geq f(x_k) + \langle s - x_k, \nabla f(x_k) \rangle = \hat{f}_k(s)$$

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

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

The right hand side of this inequality is readily computed after step (2).

3.1 CGM for sparse inverse problems

In this section we translate the classical CGM for the sparse inverse problem (1.3). We give two versions—first a direct translation of the fully corrective variant and then our improved algorithm for differentiable measurement models. To make it clear that we operate over the space of measures we change notation and denote the iterate by μ_k instead of x_k . The most obvious challenge is that we cannot represent a general measure on a computer unless it is finitely-supported. We will see however that the steps of CGM can in fact be carried out on a computer in this context. Moreover we later prove that the iterates can be represented with bounded memory.

Before we describe the algorithm in detail, we first explain how to linearize the objective and minimize the linearization. In the space of measures, linearization is most easily understood as a directional derivative: in the finite dimensional case, we always have that

$$\langle \nabla f(\mu_k), s \rangle = D_s f(\mu_k) := \lim_{t \downarrow 0} \frac{f(\mu_k + ts) - f(\mu_k)}{t}.$$

In our formulation (1.3), $f(\mu) = \ell(\Phi\mu - y)$. If we define the *residual error* as $r_k = \Phi\mu_k - y$, we can compute the directional derivative of our particular choice of f at μ_k as

$$D_s f(\mu_k) = \lim_{t \downarrow 0} \frac{\ell(\Phi(\mu_k + ts) - y) - \ell(\Phi\mu_k - y)}{t} = \lim_{t \downarrow 0} \frac{\ell(r_k + t\Phi s) - \ell(r_k)}{t} = D_{\Phi s} \ell(r_k) = \langle \nabla \ell(r_k), \Phi s \rangle. \quad (3.2)$$

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 nonnegative measure s with total mass bounded by τ . Interchanging the integral (in Φ) with the inner product, and defining $F(\theta) := \langle \nabla \ell(r_k), \psi(\theta) \rangle$, we need to solve the optimization problem:

$$\underset{s \geq 0, s(\Theta) \leq \tau}{\text{minimize}} \int F(\theta) ds(\theta). \quad (3.3)$$

The optimal solution of (3.3) is the point-mass $\tau \delta_{\theta_*}$, where $\theta_* \in \arg \min F(\theta)$ (unless $F(\theta)$ is positive everywhere in which case the optimal solution is the 0 measure). This means that at each step of the CGM we need only add a single point to the support of our approximate solution μ_k . Moreover we prove that our algorithm produces iterates μ_k with support on at most $d + 1$ points (see Theorem 5.1).

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 a nonnegative atomic measure μ_k supported on a finite set S_k with

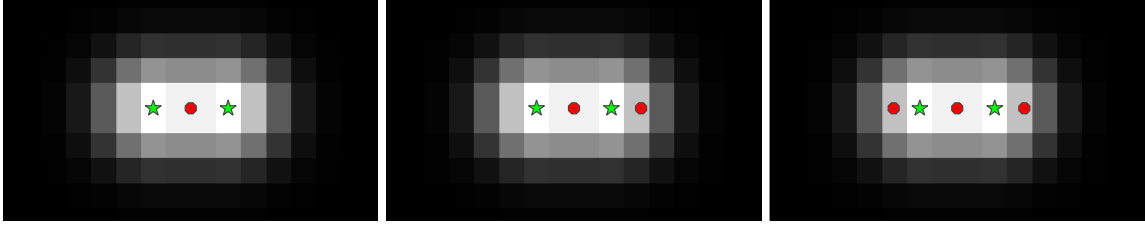


Figure 1: The three plots above show the first three iterates of the fully corrective CGM in a simulated superresolution imaging problem with two point sources of light. The locations of the true point sources are indicated by green stars, and the greyscale background shows the pixelated image. The elements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

mass $\mu_k(\{\theta\})$ on points $\theta \in S_k$. The algorithm alternates between selecting a source to add to the support, and tuning the weights to lower the current cost. This tuning step (Step 4) is a finite-dimensional convex optimization problem that we can solve with an off-the-shelf algorithm.

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

For $k = 1 : k_{\max}$

1. Compute gradient of loss: $g_k = \nabla \ell(\Phi \mu_{k-1} - y)$.
 2. Compute next source: $\theta_k \in \arg \min_{\theta \in \Theta} \langle g_k, \psi(\theta) \rangle$.
 3. Update support: $S_k \leftarrow S_{k-1} \cup \{\theta_k\}$.
 4. Compute weights: $\mu_k \leftarrow \arg \min_{\substack{\mu \geq 0, \mu(S_k) \leq \tau \\ \mu(S_k^c) = 0}} \ell(\sum_{\theta \in S_k} \mu(\{\theta\}) \psi(\theta) - y)$.
 5. Prune support: $S_k \leftarrow \text{supp}(\mu_k)$.
-

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

$$\langle \Phi \mu_k, g_k \rangle - \tau \langle \psi(\theta_k), g_k \rangle_+ < \epsilon. \quad (3.4)$$

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 Θ . Figure 1 shows CGM-M applied to an image of two closely separated sources. The first source θ_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 Θ . The following algorithm, which we call the alternating descent conditional gradient method (ADCG), exploits the differentiability of ψ 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: $g_k \leftarrow \nabla \ell(\Phi \mu_{k-1} - y)$.
 2. Compute next source: Choose $\theta_k \in \arg \min_{\theta \in \Theta} \langle \psi(\theta), g_k \rangle$.
 3. Update support: $S_k \leftarrow S_{k-1} \cup \{\theta_k\}$.
 4. Coordinate descent on nonconvex objective:
Repeat:
 - (a) Compute weights: $\mu_k \leftarrow \arg \min_{\substack{\mu \geq 0, \mu(S_k) \leq \tau \\ \mu(S_k^c) = 0}} \ell(\sum_{\theta \in S_k} \mu(\{\theta\}) \psi(\theta) - y)$.
 - (b) Prune support: $S_k = \text{support}(\mu_k)$.
 - (c) Locally improve support: $S_k = \mathbf{local_descent}((\theta, \mu_k(\{\theta\})) : \theta \in S_k)$.
-

Here **local_descent** is a subroutine that takes a measure μ_k with atomic representation $(\theta_1, w_1), \dots, (\theta_m, w_m)$ and attempts to use gradient information to reduce the function

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

holding the weights fixed.

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

$$\begin{aligned} & \text{minimize} && \ell\left(\sum_{i=1}^k w_i \psi(\theta_i) - y\right) \\ & \text{subject to} && w_i \geq 0 \\ & && \theta_i \in \Theta \\ & && \sum_i w_i \leq \tau \end{aligned} \tag{3.5}$$

is nonconvex. Step 4 is then block coordinate descent over w_i and θ_i . The algorithm as a whole can be interpreted as alternating between performing descent on the convex (but infinite-dimensional) problem (1.3) 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.

As we have previously discussed, this nonconvex local search does not change the convergence guarantees of the CGM whatsoever. We will show in Section 5 that this is an immediate consequence of the existing theory on the CGM. However, as we will show in §6, the inclusion of this local search dramatically improves the performance of the CGM.

3.2 Interface and implementation

Roughly speaking, running ADCG on a concrete instance of (1.3) requires subroutines for two operations. We need algorithms to approximately compute:

- (a) $\psi(\theta)$ and $\frac{d}{d\theta} \psi(\theta)$ for $\theta \in \Theta$.
- (b) $\arg \min_{\theta \in \Theta} \langle \psi(\theta), v \rangle$ 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 Θ 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 Section 5.

If the parameter space is higher 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 [25, 38, 20, 49]. 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, Θ 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 conditional gradient method as a general purpose solver for constrained inverse problems [24, 20]. 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 classic conditional gradient method 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 [24]. Moreover, Rao *et al* [38] have proposed a version of the conditional gradient method, called CoGENT, for atomic norm problems that take 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 [49, 30, 20, 25] have proposed algorithms based on combinations of rank-one updates and local nonconvex optimization inspired by the well-known heuristic of [8]. 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 (1.3). Tang *et al* [46] 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 even of relatively modest dimension. Moreover, even when gridding is tractable, the solutions obtained are often supported on very large sets and heuristic post-processing is required to achieve reasonable performance in practice [46]. In spite of these limitations, gridding is the state of the art in many application areas including computational neuroscience [15], superresolution fluorescence microscopy [50], radar [5, 21], remote sensing [17], compressive sensing [4, 12, 13], and polynomial interpolation [39].

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 [7] propose an algorithm to solve the Tikhonov-regularized version of problem (1.3) 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. For the same reason, they are limited to one pass of gradient descent in the nonconvex phase of the algorithm. In §6 we show that this limitation has serious performance implications in practice.

5 Theoretical guarantees

In this section we present a few theoretical results. The first guarantees that we can run our algorithm with bounded memory. The second result guarantees that the algorithm converges to an optimal point and bounds the worst-case rate of convergence.

5.1 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 μ_k is bounded by k . For large k , then, μ_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 5.1. *ADCG may be implemented to generate iterates with cardinality of support uniformly bounded by $d + 1$.*

Proof. Lemma (5.2) allows us to conclude that the fully-corrective step ensures that the support of the measure remains bounded by $d + 1$ for all iterations. \square

Lemma 5.2. *The finite-dimensional problem*

$$\underset{w \geq 0, \sum_{i=1}^m w_i \leq \tau}{\text{minimize}} \quad \ell\left(\sum_i w_i \psi(\theta_i) - y\right) \quad (5.1)$$

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

Proof. Let u_* be any optimal solution to (5.1). As u_* is feasible, we have that

$$v = \sum_i u_{*i} \psi(\theta_i) \in \tau \text{conv}(\{\psi(\theta_i) : i = 1, \dots, m\} \cup \{0\}).$$

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 $\{\psi(\theta_i) : i = 1, \dots, m\}$. That is, there exists a w_* with at most $d + 1$ nonzeros such that

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

This implies that w_* is also optimal for (5.1). \square

Note that in order to find w_* , we need to either use a simplex-type algorithm to solve (5.1) or explore the optimal set using the random ray-shooting procedure as described in [45].

5.2 Convergence analysis

We now analyze the worst-case convergence rate for ADCG applied to (1.3). Theorem 5.3 below guarantees that ADCG achieves accuracy δ in $\mathcal{O}(\frac{1}{\delta})$ iterations.

The theorem applies even when the linear minimization step is performed approximately. That is, we allow θ_k to be chosen such that

$$\langle \psi(\theta_k), g_k \rangle \leq \min_{\theta \in \Theta} \langle \psi(\theta), g_k \rangle + \frac{\zeta}{k + 2} \quad (5.2)$$

for some $\zeta \geq 0$. When inequality (5.2) holds, we say that the linear minimization problem in iteration k is solved to precision ζ .

The analysis relies on a finite-dimensional optimization problem equivalent to (1.3). Let $\mathcal{A} = \{\psi(\theta) : \theta \in \Theta\} \cup \{0\}$. Readers familiar with the literature on atomic norms [11] will recognize the finite-dimensional problem we consider as an atomic norm problem:

$$\begin{aligned} & \underset{x \in \mathbb{R}^d}{\text{minimize}} && \ell(x - y) \\ & \text{subject to} && x \in \tau \text{conv} \mathcal{A}. \end{aligned} \tag{5.3}$$

The connection to (1.3) becomes clear if we note that $\tau \text{conv} \mathcal{A} = \{\Phi\mu : \mu \geq 0, \mu(\Theta) \leq \tau\}$. Any feasible measure μ for (1.3) gives us a feasible point $\Phi\mu$ for (5.3). Likewise, any feasible x for (5.3) can be decomposed as a feasible measure μ for (1.3). Furthermore, these equivalences preserve the objective value.

Before we state the theorem precisely, we introduce some notation. Let $\ell_\star = \ell(\Phi\mu_\star - y)$ denote the optimal value of (1.3)—the discussion above implies that ℓ_\star is also the optimal value of (5.3). Following Jaggi in [24], we define the curvature parameter $C_{f,\mathcal{S}}$ of a function f on a set \mathcal{S} . Intuitively, $C_{f,\mathcal{S}}$ measures the maximum divergence between f and its first-order approximations, $\hat{f}(z; x) = f(x) + \langle z - x, \nabla f(x) \rangle$:

$$C_{f,\mathcal{S}} = \sup_{\substack{x, s \in \mathcal{S} \\ \gamma \in [0, 1] \\ z = x + \gamma(s - x)}} \frac{2}{\gamma^2} (f(z) - \hat{f}(z; x)).$$

Theorem 5.3. *Let C be the curvature parameter of the function $f(x) = \ell(x - y)$ on the set $\tau \text{conv} \mathcal{A}$. If each linear minimization subproblem is solved to precision $C\zeta$, the iterates μ_1, μ_2, \dots of ADCG applied to (1.3) satisfy*

$$\ell(\Phi\mu_k - y) - \ell_\star \leq \frac{2C}{k+2} (1 + \zeta).$$

Proof. We first show that the points $\Phi\mu_1, \Phi\mu_2, \dots$ are iterates of the standard CGM (with a particular choice of the final update step) applied to the finite-dimensional problem (5.3). We then appeal to [24] to complete the proof.

Suppose that $\Phi\mu_k = x_k$. We show that the linearization step in both algorithms produces the same result (up to the equivalence mentioned earlier). Let $\theta_{k+1} = \arg \min_{\theta \in \Theta} \langle \psi(\theta), \nabla \ell(\Phi\mu_k - y) \rangle$ be the output of step 2 of ADCG. Let s_k be the output of the linear minimization step of the standard CGM applied to (5.3) starting at x_k . Then

$$s_k = \arg \min_{s \in \tau \text{conv} \mathcal{A}} \langle s, \nabla \ell(x_k - y) \rangle.$$

Recalling that $\text{conv} \mathcal{A} = \{\Phi\mu \mid \mu \geq 0, \mu(\Theta) \leq 1\}$, we must have $s_k = \tau\psi(\theta_k)$. Therefore, the linear minimization steps of the standard CGM and ADCG coincide.

We now need to show that the nonconvex coordinate descent step in ADCG is a valid final update step for the standard CGM applied to (5.3). This is clear as the coordinate descent step does at least as well as the fully-corrective step. We can hence appeal to the results of Jaggi [24] that bound the convergence rate of the standard CGM on finite-dimensional problems to finish the proof. \square

6 Numerical results

In this section we apply ADCG to three of the examples in §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 (1.3).

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 conditional gradient method for measures (CGM-M) and a variant of the gradient flow algorithm (GF) proposed in [7]. While the gradient flow algorithm proposed in [7] 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 τ by inspection. For matrix completion and linear system ID this means using a validation set. For single molecule imaging each image requires a different value of τ . For this problem, we run ADCG with a large value of τ 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 τ and the stopping tolerance ϵ , or as a stagewise algorithm [48].

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 (1.3). For this reason we do not include detailed timing information.

6.1 Superresolution fluorescence microscopy

We analyze data from the Single Molecule Localization Microscopy (SMLM) challenge [42, 18]. Fluorescence microscopy is an imaging technique used in the biological sciences to study subcellular structures in vivo. The task is to recover the 2D positions of a collection of fluorescent proteins from images taken through an optical microscope.

Here we compare the performance of our ADCG to the gridding approach of Tang *et al* [46], two algorithms from the microscopy community (quickPALM and center of Gaussians), and also CGM and the gradient flow (GF) algorithm proposed by [7]. The gridding approach approximately solves the continuous optimization problem (1.3) by discretizing the space Θ into a finite grid of candidate point source locations and running an ℓ_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 *et al* propose a heuristic post-processing step that involves taking the center of mass of each cluster. This post-processing step is hard to understand theoretically, and does not perform well with a high-density of fluorophores.

6.1.1 Implementation details

For this application, the minimization required in step 2 of ADCG is not difficult: the parameter space is two-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 [27].

6.1.2 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 [18] 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 [18]. There are 81049 point sources in total, roughly evenly distributed across the images. Figure 2a shows a typical image. Each image covers an area 6400nm across, meaning each pixel is roughly 100nm by 100nm.

Figure 3 compares the performance of ADCG, gridding, quickPALM, and center of Gaussians (CoG) on this dataset. We match the performance of the gridding algorithm from [46], and significantly beat both quickPALM and CoG. Our algorithm analyses all images in well under an hour—significantly faster than the

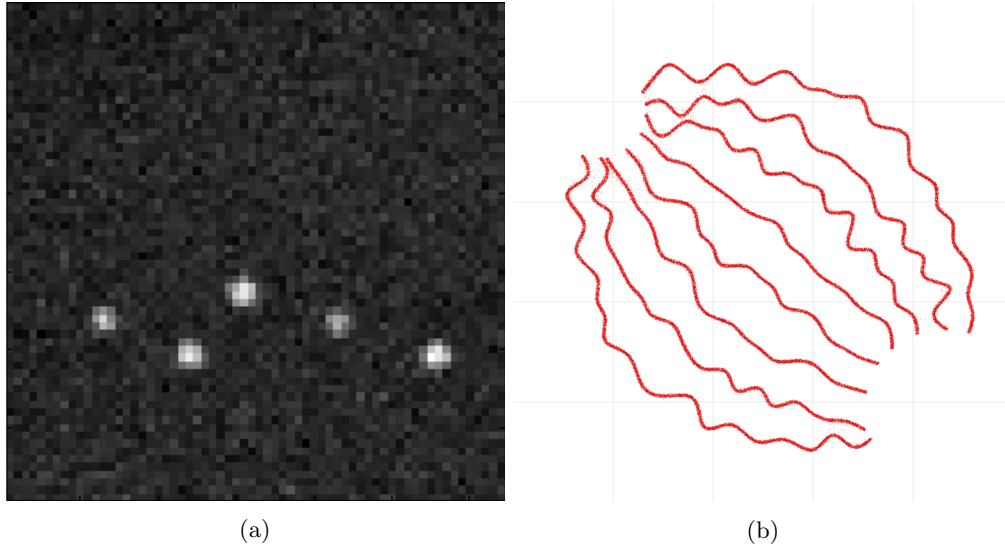


Figure 2: The long sequence dataset contains 12000 images similar to (a). The recovered locations for all the images are displayed in (b).

gridding approach of [46]. Note that the gridding algorithm of [46] does not work without a post-processing step.

6.2 Matrix completion

As described in §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.

6.2.1 Implementation details

Although the parameter space for this example is high-dimensional we can still compute the steepest descent step over the space of measures. The optimization problem we need to solve is:

$$\underset{\|a\|_2=\|b\|_2=1}{\text{minimize}} \langle \psi(a, b), \nu \rangle = \langle M(ab^T), \nu \rangle = \langle ab^T, M^*(\nu) \rangle.$$

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 with line-search.

6.2.2 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 480,189 Netflix users give to a subset of 17,770 movies. One approach has been to phrase this as a matrix completion problem. That is, to try to complete the 480,189 by 17,770 matrix of ratings from the observed entries. Following [40] 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, AD CG outperforms all other CGM variants. Our algorithm takes over 7

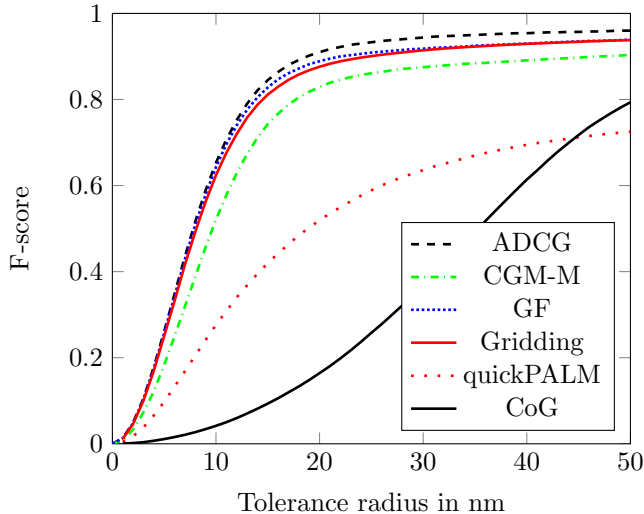


Figure 3: **Performance on bundled tubes: long sequence.** F-scores at various radii for 6 algorithms. For reference, each image is 6400nm across, meaning each pixel has a width of 100nm. ADCG outperforms all competing methods on this dataset.

hours to achieve the best RMSE—this could be improved with a more sophisticated implementation, or parallelization.

6.3 System identification

In this section we apply our algorithms to identifying two single-input single-output systems from the DaISy collection [14]: the flexible robot arm dataset (ID 96.009) and the hairdryer dataset (ID 96.006).

6.3.1 Implementation details

While the parameter space is 6-dimensional, which effectively precludes gridding, we can efficiently solve the minimization problem in step (2) of the ADCG. To do this, we grid only over r and α : 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 , α .

For `local_descent` we again use a standard box-constrained gradient method provided by the NLOpt library [27].

6.3.2 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 [43] 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_{pred} using the score defined in [19]. The score is given by

$$\text{score} = 100 \left(1 - \frac{\|y_{\text{pred}} - y\|_2}{\|y_{\text{mean}} - y\|_2} \right), \quad (6.1)$$

where y_{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 [19]. ADCG matches the performance of [19] and exceeds that of all

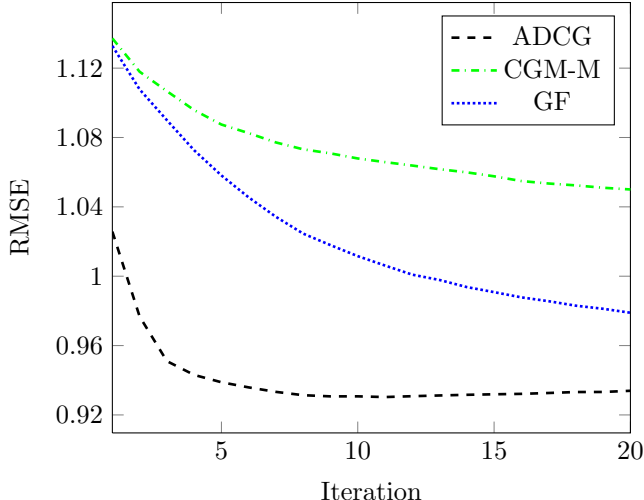


Figure 4: **RMSE on Netflix challenge dataset.** ADCG significantly outperforms CGM-M.

other CGM variants. Our simple implementation takes about an hour, which compares very poorly with the spectral methods in [19] which complete in under a minute.

7 Conclusions and future work

As demonstrated in the numerical experiments of §6, ADCG achieves state of the art performance in superresolution fluorescence microscopy, matrix completion, and system identification, without the need for heuristic post-processing 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 conditional gradient method 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 conditional gradient method, 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 [28, 26], sparse coding [1, 2], and phase retrieval [33], and perhaps the techniques from this area could be applied to our setting of sparse inverse problems.

Relaxation 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, 34]. 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

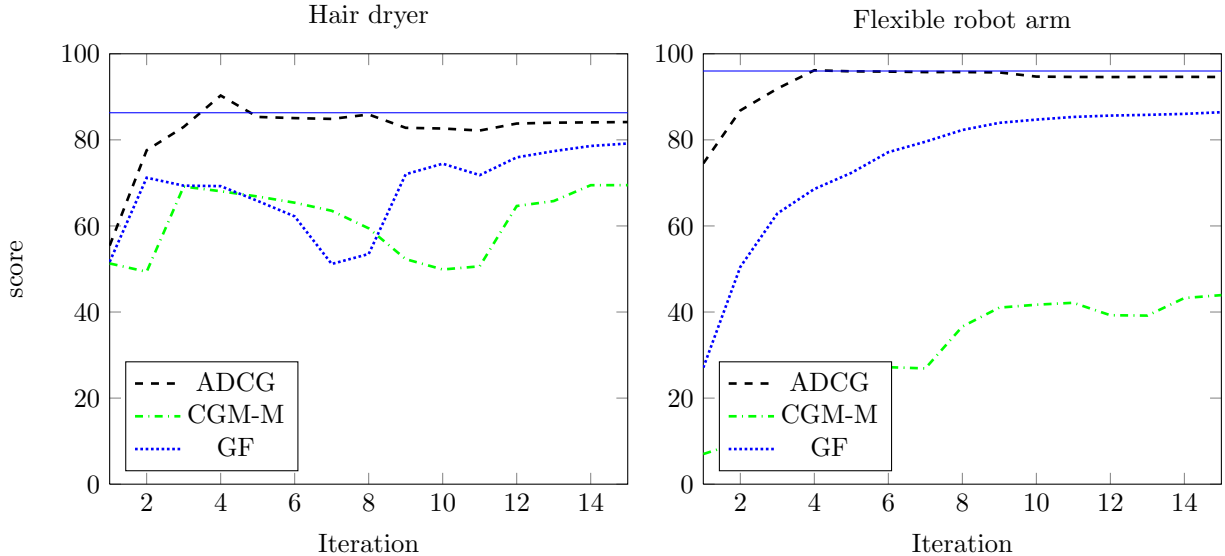


Figure 5: **Performance on DaISy datasets.** ADCG outperforms other CGM variants and matches the nuclear-norm based technique of [43].

global optimal solution [34]. It is possible 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 (1.3) is a semi-infinite program (SIP), namely an optimization problem with a finite dimensional decision variable but an infinite collection of constraints [22, 44]. 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 semi-infinite programs 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 §2, we have made our code publicly available on GitHub. As described in §3, implementing ADCG for a new application essentially requires only two user-specified subroutines: one routine that evaluates the the measurement 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.

Acknowledgements

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

BR is 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. GS was generously supported by NSF award CCF-1148243. NB 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.

References

- [1] A. Agarwal et al. “Learning sparsely used overcomplete dictionaries via alternating minimization”. In: *arXiv preprint arXiv:1310.7991* (2013).
- [2] S. Arora et al. “Simple, efficient, and neural algorithms for sparse coding”. In: *arXiv 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*. 2007, pp. 1027–1035.
- [4] W. Bajwa et al. “Compressed channel sensing: A new approach to estimating sparse multipath channels”. In: *Proc. IEEE* 98.6 (2010), pp. 1058–1076.
- [5] R. Baraniuk and P. Steeghs. “Compressive radar imaging”. In: *In IEEE Radar Conf., Waltham, MA* (2007), pp. 128–133.
- [6] E. Van den Berg and M. P. Friedlander. “Sparse optimization with least-squares constraints”. In: *SIAM Journal on Optimization* 21.4 (2011), pp. 1201–1229.
- [7] K. Bredies and H. K. Pikkarainen. “Inverse problems in spaces of measures”. In: *ESAIM: Control, Optimisation and Calculus of Variations* 19 (01 Jan. 2013), pp. 190–218. ISSN: 1262-3377. DOI: [10.1051/cocv/2011205](https://doi.org/10.1051/cocv/2011205). URL: http://www.esaim-cocv.org/article_S1292811911002053.
- [8] S. Burer and R. D. Monteiro. “A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization”. English. In: *Mathematical Programming* 95.2 (2003), pp. 329–357. ISSN: 0025-5610. DOI: [10.1007/s10107-002-0352-8](https://doi.org/10.1007/s10107-002-0352-8). URL: <http://dx.doi.org/10.1007/s10107-002-0352-8>.
- [9] E. Candes and C. Fernandez-Granda. “Towards a mathematical theory of super resolution”. In: *Comm. Pure Appl. Math* (2013).
- [10] E. Candès and B. Recht. “Exact matrix completion via convex optimization”. In: *Communications of the ACM* 55.6 (2012), pp. 111–119.
- [11] V. Chandrasekaran et al. “The Convex Geometry of Linear Inverse Problems.” In: *Foundations of Computational Mathematics* 12.6 (2012), pp. 805–849. URL: <http://dblp.uni-trier.de/db/journals/focm/focm12.html#ChandrasekaranRPW12>.
- [12] M. C. D. Malioutov and A. Willsky. “A sparse signal reconstruction perspective for source localization with sensor arrays”. In: *IEEE Trans. Signal Process* 53.8 (2005), pp. 3010–3022.
- [13] M. Duarte and R. Baraniuk. “Spectral compressive sensing”. In: *Applied and Computational Harmonic Analysis* 35.1 (2013), pp. 111–129.
- [14] D. M. B. (ed.) *DaISy: Database for the Identification of Systems*, URL: <http://homes.esat.kuleuven.be/~smc/daisy/>.
- [15] C. Ekanadham, D. Tranchina, and E. P. Simoncelli. “Neural spike identification with continuous basis pursuit”. In: *Computational and Systems Neuroscience (CoSyNe), Salt Lake City, Utah* (2011).
- [16] D. Evanko. “Primer: fluorescence imaging under the diffraction limit”. In: *Nature Methods* 6 (2009), pp. 19–20.
- [17] A. Fannjiang, T. Strohmer, and P. Yan. “Compressed remote sensing of sparse objects”. In: *SIAM J. Imag. Sci.* 3.3 (2010), pp. 595–618.
- [18] B. I. Group. *Benchmarking of Single-Molecule Localization Microscopy Software*. 2013. URL: <http://bigwww.epfl.ch/palm/>.

- [19] A. Hansson, Z. Liu, and L. Vandenberghe. “Subspace System Identification via Weighted Nuclear Norm Optimization”. In: *CoRR* abs/1207.0023 (2012). URL: <http://arxiv.org/abs/1207.0023>.
- [20] Z. Harchaoui, A. Juditsky, and A. Nemirovski. “Conditional gradient algorithms for norm-regularized smooth convex optimization”. In: *Mathematical Programming* (2014), pp. 1–38.
- [21] M. Herman and T. Strohmer. “High-resolution radar via compressed sensing”. In: *IEEE Trans. Signal Process.* 57.6 (2009), pp. 2275–2284.
- [22] R. Hettich and K. O. Kortanek. “Semi-infinite programming: theory, methods, and applications”. In: *SIAM review* 35.3 (1993), pp. 380–429.
- [23] H. Hindi. “A tutorial on optimization methods for cancer radiation treatment planning”. In: *American Control Conference (ACC), 2013.* 2013, pp. 6804–6816. DOI: [10.1109/ACC.2013.6580908](https://doi.org/10.1109/ACC.2013.6580908).
- [24] M. Jaggi. “Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization”. In: *ICML* (2013).
- [25] M. Jaggi, M. Sulovsk, et al. “A simple algorithm for nuclear norm regularized problems”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10).* 2010, pp. 471–478.
- [26] 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. 2013, pp. 665–674.
- [27] S. G. Johnson. *The NLOpt nonlinear-optimization package.* 2011. URL: <http://ab-initio.mit.edu/nlopt>.
- [28] R. H. Keshavan. “Efficient algorithms for collaborative filtering”. PhD thesis. Stanford University, 2012.
- [29] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.* The MIT Press, 2009. ISBN: 0262013193, 9780262013192.
- [30] S. Laue. “A Hybrid Algorithm for Convex Semidefinite Optimization”. In: *Proceedings of the 29th International Conference on Machine Learning (ICML-12).* Ed. by J. Langford and J. Pineau. ICML ’12. Edinburgh, Scotland, GB: Omnipress, 2012, pp. 177–184. ISBN: 978-1-4503-1285-1.
- [31] H.-Y. Liu et al. “3D imaging in volumetric scattering media using phase-space measurements”. In: *Opt. Express* 23.11 (2015), pp. 14461–14471. DOI: [10.1364/OE.23.014461](https://doi.org/10.1364/OE.23.014461). URL: <http://www.opticsexpress.org/abstract.cfm?URI=oe-23-11-14461>.
- [32] D. Malioutov, M. Çetin, and A. S. Willsky. “A sparse signal reconstruction perspective for source localization with sensor arrays”. In: *Signal Processing, IEEE Transactions on* 53.8 (2005), pp. 3010–3022.
- [33] P. Netrapalli, P. Jain, and S. Sanghavi. “Phase retrieval using alternating minimization”. In: *Advances in Neural Information Processing Systems.* 2013, pp. 2796–2804.
- [34] R. Ostrovsky et al. “The Effectiveness of Lloyd-Type Methods for the k -Means Problem”. In: *Journal of the ACM* 59.6 (2012), 28:1–28:22.
- [35] 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”. In: *Journal de l’ecole Polytechnique* 1.22 (1795), pp. 24–76.
- [36] F. Pukelsheim. *Optimal design of experiments.* Vol. 50. siam, 1993.
- [37] K. G. Puschmann and F. Kneer. “On super-resolution in astronomical imaging”. In: *Astronomy and Astrophysics* 436 (2005), pp. 373–378.
- [38] N. Rao, P. Shah, and S. Wright. “Forward-Backward Greedy Algorithms for Atomic Norm Regularization”. In: *arXiv:1404.5692* (2014).
- [39] H. Rauhut. “Random sampling of sparse trigonometric polynomials”. In: *Applied and Computational Harmonic Analysis* 22.1 (2007), pp. 16–42.

- [40] B. Recht and C. Ré. “Parallel stochastic gradient algorithms for large-scale matrix completion”. In: *Mathematical Programming Computation* 5.2 (2013), pp. 201–226.
- [41] M. Rust, M. Bates, and X. Zhuang. “Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)”. In: *Nature Methods* 3 (2006), pp. 793–796.
- [42] D. Sage et al. “Quantitative evaluation of software packages for single-molecule localization microscopy”. In: *Nat Meth* advance online publication (June 2015), pp. –. URL: <http://dx.doi.org/10.1038/nmeth.3442>.
- [43] P. Shah et al. “Linear System Identification via Atomic Norm Regularization”. In: *arXiv:1204.0590* (2012).
- [44] A. Shapiro. “Semi-infinite programming, duality, discretization and optimality conditions”. In: *Optimization* 58.2 (2009), pp. 133–161.
- [45] J. Skaf and S. Boyd. “Techniques for exploring the suboptimal set”. In: *Optimization and Engineering* 11.2 (2010), pp. 319–337.
- [46] G. Tang, B. Bhaskar, and B. Recht. “Sparse recovery over continuous dictionaries: Just discretize”. In: *Asilomar* (2013).
- [47] G. Tang et al. “Compressed sensing off the grid”. In: *IEEE Trans. Inf. Thy* 59.11 (2013), pp. 7465–7490.
- [48] R. J. Tibshirani. “A General Framework for Fast Stagewise Algorithms”. In: *arXiv preprint arXiv:1408.5801* (2014).
- [49] X. Zhang, Y. Yu, and D. Schuurmans. “Accelerated Training for Matrix-norm Regularization: A Boosting Approach”. In: *Advances in Neural Information Processing Systems 26 (NIPS)*. 2012.
- [50] L. Zhu et al. “Faster storm using compressed sensing”. In: *Nature Methods* 9 (2012), pp. 721–723.