SPARSE INVERSE PROBLEMS OVER MEASURES: EQUIVALENCE OF THE CONDITIONAL GRADIENT AND EXCHANGE METHODS

ARMIN EFTEKHARI AND ANDREW THOMPSON*

Abstract. We study an optimization program over nonnegative Borel measures that encourages sparsity in its solution. Efficient solvers for this program are in increasing demand, as it arises when learning from data generated by a "continuum-of-subspaces" model, a recent trend with applications in signal processing, machine learning, and high-dimensional statistics. We prove that the conditional gradient method (CGM) applied to this infinite-dimensional program, as proposed recently in the literature, is equivalent to the exchange method (EM) applied to its Lagrangian dual, which is a semi-infinite program. In doing so, we formally connect such infinite-dimensional programs to the well-established field of semi-infinite programming.

On the one hand, the equivalence established in this paper allows us to provide a rate of convergence for EM which is more general than those existing in the literature. On the other hand, this connection and the resulting geometric insights might in the future lead to the design of improved variants of CGM for infinite-dimensional programs, which has been an active research topic. CGM is also known as the Frank-Wolfe algorithm.

1. Introduction. We consider the following affinely-constrained optimization over nonnegative Borel measures:

$$(1.1) \qquad \begin{cases} \min\limits_{x} & L\left(\displaystyle\int_{\mathbb{I}} \Phi(t)x(dt) - y\right) \\ \text{subject to} & \|x\|_{TV} \le 1 \\ & x \in B_+(\mathbb{I}). \end{cases}$$

Here, $\mathbb{I}$ is a compact subset of Euclidean space, $B_+(\mathbb{I})$ denotes all nonnegative Borel measures supported on $\mathbb{I}$, and

$$(1.2) \qquad \|x\|_{TV} = \int_{\mathbb{I}} x(dt)$$

is the *total variation* of measure x, see for example [1].[1] We are particularly interested in the case where $L : \mathbb{C}^m \to \mathbb{R}$ is a differentiable *loss function* and $\Phi : \mathbb{I} \to \mathbb{C}^m$ is a continuous function. Note that Program (1.1) is an *infinite-dimensional* problem and that the constraints ensure that the problem is bounded. In words, Program (1.1) searches for a nonnegative measure on $\mathbb{I}$ that minimizes the loss above, while controlling its total variation. This problem and its variants have received significant attention [3, 4, 5, 6, 7, 8, 9] in signal processing and machine learning, see Section 2 for more details.

It was recently proposed in [4] to solve Program (1.1) using the celebrated *conditional gradient method* (CGM) [10], also known as the Frank-Wolfe algorithm, adapted to optimization over nonnegative Borel measures. The CGM algorithm minimizes a differentiable, convex function over a compact convex set, and proceeds by iteratively minimizing linearizations of the objective function over the feasible set, generating a new descent direction in each iteration. The classical algorithm performs a descent step in each new direction generated, while in the *fully-corrective* CGM, the objective is minimized over the subspace spanned by all previous directions [11]. It is the fully-corrective version of the algorithm which we consider in this paper.

It was shown in [4] that, when applied to Program (1.1), CGM generates a sequence of finitely supported measures, with a single parameter value $t^l \in \mathbb{I}$ being added to the support in the lth iteration. Moreover, [4] established that the convergence rate of CGM here is $\mathcal{O}\left(\frac{1}{l}\right)$, where l is the number of iterations, thereby extending the standard results for finite-dimensional CGM. A full description of CGM and its convergence guarantees can be found in Section 3.

On the other hand, the (Lagrangian) dual of Program (1.1) is a finite-dimensional optimization problem with infinitely many constraints, often referred to as a *semi-infinite program* (SIP), namely

$$(1.3) \qquad \begin{cases} \max\limits_{\lambda,\alpha} & \operatorname{Re}\langle\lambda, y\rangle - L_\circ(-\lambda) - \alpha \\ \text{subject to} & \operatorname{Re}\langle\lambda, \Phi(t)\rangle \le \alpha, \qquad t \in \mathbb{I} \\ & \alpha \ge 0, \end{cases}$$

---

*AE and AT have contributed equally to this work. AE is with the Institute of Electrical Engineering at the École Polytechnique Fédérale de Lausanne, Switzerland. AT is with the National Physical Laboratory, United Kingdom.
[1] It is also common to define the TV norm as half of the right-hand side of (1.2), see [2].

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product over $\mathbb{C}^m$. Above,

$$(1.4) \qquad L_\circ(\lambda) = \sup_{z \in \mathbb{C}^m} \mathrm{Re}\langle \lambda, z \rangle - L(z)$$

denotes the Fenchel conjugate of L. As an example, when $L(\cdot) = \frac{1}{2} \| \cdot \|_2^2$, it is easy to verify that $L_\circ = L$. For the sake of completeness, we verify the duality of Programs (1.1) and (1.3) in Appendix B. Note that the Slater's condition for the finite-dimensional Program (1.3) is met and there is consequently no duality gap between the two Programs (1.1) and (1.3).

There is a large body of research on SIPs such as Program (1.3), see for example [12, 13, 14], and we are particularly interested in solving Program (1.3) with *exchange methods*. In one instantiation – which for ease we will refer to as *the* exchange method (EM) – one forms a sequence of nested subsets of the constraints in Program (1.3), adding in the lth iteration a single new constraint corresponding to the parameter value $t^l \in \mathbb{I}$ that maximally violates the constraints of Program (1.3). The finite-dimensional problem with these constraints is then solved and the process repeated. Convergence of EM has been established under somewhat general conditions, but results concerning rate of convergence are restricted to more specific SIPs, see Section 4 for a full description of the EM.

Contribution. The main contribution of this paper is to establish that, for Program (1.1) and provided the loss function L is both strongly smooth and strongly convex, *CGM and EM are dual-equivalent.* More precisely, the iterates of the two algorithms produce the same objective value and the same finite set of parameters in each iteration; for CGM, this set is the support of the current iterate of CGM and, for EM, this set is the choice of constraints in the dual program.

The EM method can also be viewed as a *bundle method* for Program (1.3) as discussed in Section 6, and the duality of CGM and bundle methods is well known for finite-dimensional problems. This paper establishes dual-equivalence in the emerging context of optimization over measures on the one hand and the well-established semi-infinite programming on the other hand.

On the one hand, the equivalence established in this paper allows us to provide a rate of convergence for EM which is more general than those existing in the literature; see Section 6 for a thorough discussion of the prior art. On the other hand, this connection and the resulting geometric insights might lead to the design of improved variants to CGM, another active research topic [4].

Outline. We begin in Section 2 with some motivation, describing the key role of Program (1.1) in data and computational sciences. Then in Sections 3 and 4, we give a more technical introduction to CGM and EM, respectively. We present the main contributions of the paper in Section 5, establishing the dual-equivalence of CGM and EM for Problems (1.1) and (1.3), and deriving the rate of convergence for EM. Related work is reviewed in Section 6 and some geometric insights into the inner workings of CGM and EM are provided in Section 7. We conclude this paper with a discussion of the future research directions.

2. Motivation. Program (1.1) has diverse applications in data and computational sciences. In signal processing for example, each $\Phi(t) \in \mathbb{C}^m$ is an *atom* and the set of all atoms $\{\Phi(t)\}_{t \in \mathbb{I}}$ is sometimes referred to as the *dictionary*. In radar applications, for instance, $\Phi(t)$ is a copy of a known *template*, arriving at time t. In this context, we are interested in *signals* that have a *sparse* representation in this dictionary, namely signals that can be written as the superposition of a small number of atoms. Any such signal $\dot{y} \in \mathbb{C}^m$ can be written as

$$(2.1) \qquad \dot{y} = \int_{\mathbb{I}} \Phi(t) \dot{x}(dt),$$

where $\dot{x}$ is a *sparse* measure, selecting the atoms that form $\dot{y}$. More specifically,

$$(2.2) \qquad \dot{x} = \sum_{i=1}^{k} \dot{a}_i \cdot \delta_{\dot{t}_i},$$

for an integer k, positive *amplitudes* $\{\dot{a}_i\}_{i=1}^{k}$, and *parameters* $\{\dot{t}_i\}_{i=1}^{k} \subset \mathbb{I}$. Here, $\delta_{\dot{t}_i}$ is the Dirac measure located at $\dot{t}_i \in \mathbb{I}$. We can therefore rewrite (2.1) as

$$(2.3) \qquad \dot{y} = \int_{\mathbb{I}} \Phi(t) \dot{x}(dt) = \sum_{i=1}^{k} \Phi(\dot{t}_i) \cdot \dot{a}_i.$$

In words, $\{\dot{t}_i\}_i$ are the parameters that construct the signal $\dot{y}$ and $\mathbb{I}$ is the *parameter space*. We often receive $y \in \mathbb{C}^m$, a noisy copy of $\dot{y}$, and our objective in signal processing is to estimate the hidden parameters $\{\dot{t}_i\}_i$, given the noisy copy y. See Figure 2.1 for an example.

(a) (b)

Figure 2.1: In this numerical example, (a) depicts the measure $\dot{x}$, see (2.2). Let $\phi(t) = e^{-100t^2}$ be a Gaussian window. With the choice of sampling locations $\{s_j\}_{j=1}^m \subset [0,1]$ and $\Phi(t) = [\phi(t-s_j)]_{j=1}^m \in \mathbb{R}^m$, (b) depicts $\dot{y} \in \mathbb{R}^m$, see (2.3). Note that the entries of $\dot{y}$ are in fact samples of $(\phi \star x)(s) = \int_{\mathbb{I}} \phi(t-s)x(dt)$ at locations $s \in \{s_j\}_{j=1}^m$, which forms the red curve in (b). Our objective is to estimate the locations $\{\dot{t}_i\}_{i=1}^k$ from $\dot{y}$. (Given an estimate of the locations, the amplitudes $\{\dot{a}_i\}_{i=1}^k$ can also be estimated with a simple least-squares program.) This is indeed a difficult task: Even given the red curve $\phi \star \dot{x}$ (from which $\dot{y}$ is sampled), it is hard to see that there is an impulse located at $\dot{t}_3$. Solving Program (1.1) with $\|x\|_{TV} \leq b$ for large enough b uniquely recovers x, as proved in [1]. In this paper, we describe Algorithms 3.1 and 4.1 to solve Program (1.1), and establish their equivalence.

To that end, Program (1.1) searches for a nonnegative measure $\hat{x}$ supported on $\mathbb{I}$ that minimizes the loss $L(\int_{\mathbb{I}} \Phi(t)x(dt) - y)$, while encouraging its *sparsity* through the total variation constraint $\|x\|_{TV} \leq 1$. Under certain conditions on Φ and when $L = \frac{1}{2}\|\cdot\|_2^2$, a minimizer $\hat{x}$ of Program (1.1) is a robust estimate of the true measure $\dot{x}$ in the sense that $d(\hat{x}, \dot{x}) \leq c \cdot L(y - \dot{y})$ for a known factor c and in a certain metric d [1, 15, 7, 16].

The super-resolution problem outlined above is an example of learning under a "continuum-of-subspaces" model, in which data belongs to the union of infinitely many subspaces. For super-resolution in particular, each subspace corresponds to fixed locations $\{t_i\}_{i=1}^K$. This model is a natural generalization of the "union-of-subspaces" model, which is a central object in compressive sensing [17], wavelets [18], and feature selection in statistics [19], to name a few. The use of continuum-of-subspaces models is on the rise as it potentially addresses the drawbacks of the union-of-subspaces models, see for example [20]. As another application of Program (1.1), y might represent the training labels in a classification task or, in the classic moments problem, y might collect the moments of an unknown distribution. Various other examples are given in [4].

Note that Program (1.1) is an infinite-dimensional problem as the search is over all nonnegative measures supported on $\mathbb{I}$. It is common in practice to restrict the support of x to a uniform grid on $\mathbb{I}$, say $\{t_i\}_{i=1}^n \subset \mathbb{I}$, so that $x = \sum_{i=1}^n a_i \delta_{t_i}$ for nonnegative amplitudes $\{a_i\}_{i=1}^n$. Let $a \in \mathbb{R}_+^n$ be the vector formed by the amplitudes and concatenate the vectors $\{\Phi(t_i)\}_{i=1}^n \subset \mathbb{C}^m$ to form a (usually very flat) matrix $\Phi \in \mathbb{C}^{m \times n}$. Then we may rewrite Program (1.1) as

(2.4)
$$\begin{cases} \min_a & L(\Phi \cdot a - y) \\ \text{subject to} & \langle 1_n, a \rangle \leq 1 \\ & a \geq 0, \end{cases}$$

3

where $1_n \in \mathbb{R}^n$ is the vector of all ones. When $L(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ in particular, Program (2.4) reduces to the well-known *nonnegative Lasso* [21].

The first issue with the above "gridding" approach is that there is often a mismatch between the atoms $\{\Phi(\dot{t}_i)\}_{i=1}^k$ that are present in $\dot{y}$ and the atoms listed in Φ, namely $\{\Phi(t_i)\}_{i=1}^n$. As a result, $\dot{y}$ often does *not* have a sufficiently sparse representation in Φ. In the context of signal processing, this problem is known as the "frequency leakage", see Figure 2.2. Countering the frequency leakage by excessively increasing the grid size n leads to increased *coherence*, namely, increased similarity between the columns of Φ. In turn, the statistical guarantees for finite-dimensional problems (such as Program (2.4)) often deteriorate as the coherence grows [22, Section 1.2]. Loosely speaking, Program (2.4) does not decouple the optimization error from the statistical error, and this pitfall can be avoided by directly studying the infinite-dimensional Program (1.1), see [16]. Moreover, the gridding approach is only applicable when the parameter space $\mathbb{I}$ is low-dimensional (see the numerical example in Section 3), often requires post-processing [23], and might lead to numerical instability with larger grids, see Program (3.3). Lastly, the gridding approach ignores the continuous structure of $\mathbb{I}$ which, as discussed in Section 8, plays a key role in developing new optimization algorithms, see after (8.1). The moment technique [16, 24] is an alternative to gridding for a few special choices of Φ in Program (1.1).

This discussion encourages us to directly study the infinite-dimensional Program (1.1); it is this direction that is pursued in this work and in [4, 25, 26, 27, 28, 29]. Indeed, this direct approach provides a unified and rigorous framework, independent of gridding or its alternatives. In particular, the direct approach perfectly decouples the optimization error (caused by gridding, for instance) from the statistical error of Program (1.1), and matches the growing trend in statistics and signal processing that aims at providing theoretical guarantees for directly learning the underlying (continuous) parameter space $\mathbb{I}$ [30, 31, 6, 29, 24].

Figure 2.2: (a) depicts a translated Gaussian window, namely, $\phi(t - t_1) = e^{-100(t-t_1)^2}$ for translation $t_1 \in [0,1]$. Equivalently, $\phi(t - t_1) = (\phi \star \delta_{t_1})(t)$, as represented in (b). On the other hand, (c) shows the coefficients of the least-squares approximation of the translated window $\phi(t - t_1)$ in the dictionary $\{\phi(t-i/N)\}_{i=1}^N$ for $N = 66$. By comparing (b) and (c), we observe that $\phi(t-t_1)$ loses its sparse representation after gridding. See the discussion at the end of Section 2 for more details.

3. Conditional Gradient Method. In this section and the next one, we review two algorithms for solving Program (1.1). The first one is the conditional gradient method [10], a popular first-order algorithm for constrained optimization. The popularity of CGM partly stems from the fact that it is projection free, unlike projected gradient descent, for example, which requires projection onto the feasible set in every iteration.

More specifically, CGM solves the general constrained optimization problem

$$\min_{x \in \mathcal{F}} f(x)$$

where $f(x)$ is a differentiable function and $\mathcal{F}$ is a compact convex set. Given the current iterate x^{l-1}, CGM finds a search direction s^l which minimizes the linearized objective function, namely, s^l is a solution to

$$(3.1) \qquad \min_{s \in \mathcal{F}} f(x^{l-1}) + \langle s - x^{l-1}, \nabla f(x^{l-1}) \rangle$$

Note that we may remove the additive terms independent of s without changing the minimizers of Program (3.1). The classical CGM algorithm then takes a step along the direction $s^l - x^{l-1}$, namely

$$x^l = x^{l-1} + \gamma^l \cdot (s^l - x^{l-1}),$$

for some step size $\gamma^l \in (0, 1]$. In a similar spirit, fully-corrective CGM chooses x^l within the convex hull of all previous update directions [11]. To be specific, fully-corrective CGM (which we simply refer to as CGM henceforth) sets x^l to be a minimizer of

$$\begin{cases} \min & f(x) \\ \text{subject to} & x \in \text{conv}(s^1, \ldots, s^l). \end{cases}$$

In the context of sparse regression and classification, CGM is particularly appealing because it produces sparse iterates. Indeed, because the objective function in Program (3.1) is linear in s, there always exist a minimizer of Program (3.1) that is an extreme point of the feasible set $\mathcal{F}$. In our case, we have that

$$\mathcal{F} = \{ x \in B_+(\mathbb{I}) : \|x\|_{TV} \leq 1 \},$$

and any extreme point of $\mathcal{F}$ is therefore of the form δ_t with $t \in \mathbb{I}$. It follows that each iterate x^l of CGM is at most l-sparse, namely, supported on a subset of $\mathbb{I}$ of size at most l.

In light of the discussion above, CGM applied to (1.1) is summarized in Algorithm 3.1. Note that we might interpret Algorithm 3.1 as follows. Let x_p be a minimizer of Program (1.1), supported on the index set $T_p \subset \mathbb{I}$. If an *oracle* gave us the correct support T_p, we could have recovered x_p by solving Program (1.1) restricted to the support T_p rather than $\mathbb{I}$. Since we do not have access to such an oracle, at iteration l, Algorithm 3.1

    1. finds an atom $\Phi(t^l)$ that reduces the objective of Program (1.1) the most, namely an atom that is least correlated with the gradient at the current residual $\int_{\mathbb{I}} \Phi(\tau) x^{l-1}(d\tau) - y$, and then

    2. adds t^l to the support.

When $L(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ in particular, Algorithm 3.1 reduces to the well-known *orthogonal matching pursuit* (OMP) for sparse regression [32], adapted to measures.

The convergence rate of CGM has been established in [4], relying heavily upon [33], and is reviewed next for the sake of completeness. We first note that the infinite dimensional Program (1.1) has the same optimal value as the finite dimensional program

$$(3.4) \qquad \min_{z \in C_{\mathbb{I}}} L(z - y),$$

where $C_{\mathbb{I}} \subset \mathbb{C}^m$ is the convex hull of $\{\Phi(t)\}_{t \in \mathbb{I}} \cup \{0\}$, namely

$$(3.5) \qquad C_{\mathbb{I}} := \left\{ \int_{\mathbb{I}} \Phi(t) x(dt) : x \in B_+(\mathbb{I}), \|x\|_{TV} \leq 1 \right\}.$$

Algorithm 3.1 CGM for solving Program (1.1)

Input: Compact set $\mathbb{I}$, continuous function $\Phi : \mathbb{I} \to \mathbb{C}^m$, differentiable function $L : \mathbb{C}^m \to \mathbb{R}$, vector $y \in \mathbb{C}^m$, and tolerance $\eta \geq 0$.

Output: Nonnegative measure $\widehat{x}$ supported on $\mathbb{I}$.

Initialize: Set $l = 1$, $T^0 = \emptyset$, and $x^0 \equiv 0$.

While $\|\nabla L(\int_{\mathbb{I}} \Phi(\tau) x^{l-1}(d\tau) - y)\|_2 > \eta$, **do**

    1. Let t^l be a minimizer of

$$(3.2) \qquad \min_{t \in \mathbb{I}} \left\langle \Phi(t), \nabla L \left(\int_{\mathbb{I}} \Phi(\tau) x^{l-1}(d\tau) - y \right) \right\rangle.$$

    2. Set $T^l = T^{l-1} \cup \{t^l\}$.
    3. Let x^l be a minimizer of

$$(3.3) \qquad \begin{cases} \min\limits_{x} & L\left(\int_{\mathbb{I}} \Phi(t) x(dt) - y \right) \\ \text{subject to} & \|x\|_{TV} \leq 1 \\ & \text{supp}(x) \subseteq T^l \\ & x \in B_+(\mathbb{I}). \end{cases}$$

Return: $\widehat{x} = x^l$.

Indeed, both problems share the same objective value and their respective solutions $\widehat{z}$ and $\hat{x}$ satisfy

$$\widehat{z} = \int_{\mathbb{I}} \Phi(t) \widehat{x}(dt).$$

It should be emphasized that, while the problems are in this sense equivalent, solving Program (3.4) does not recover the underlying sparse measure but only its projection into the measurement space $\mathbb{C}^m$. As described in Section 2, in many applications it is precisely the underlying sparse measure which is of interest. A convergence result for CGM applied to Program 1.1 may be obtained by first establishing that its iterates x^i are related to the iterates z^i of CGM applied to the finite-dimensional Program (3.4) by $z^l = \int_{\mathbb{I}} \Phi(t) x^l(dt)$. The convergence proof from [33] can then be followed to obtain the convergence rate. Let us now turn to the details.

For the rest of this paper, we assume that L is both *strongly smooth* and *strongly convex*, namely, there exists $\gamma \geq 1$ such that

$$(3.6) \qquad \frac{\|x - x'\|_2^2}{2\gamma} \leq L(x) - L(x') - \langle x - x', \nabla L(x') \rangle \leq \frac{\gamma}{2} \|x - x'\|_2^2,$$

for every $x, x' \in \mathbb{C}^m$. In words, L can be approximated by quadratic functions at any point of its domain. For example, $L(\cdot) = \frac{1}{2}\| \cdot \|_2^2$ satisfies (3.6) with $\gamma = 1$. Let us also define

$$(3.7) \qquad r := \max_{t \in \mathbb{I}} \|\Phi(t)\|_2.$$

The convergence rate of Algorithm 3.1 is given by the following result, which is similar to the result originally given in [4], except that we replace the curvature condition in [4] with the strongly smooth and convex assumption in (3.6), see Appendix A for the proof.

PROPOSITION 1. **(Convergence rate of Algorithm 3.1)** *For $\gamma \geq 1$, suppose that L satisfies (3.6).*[2] *Suppose that Program (3.2) is solved to within an accuracy of $2\gamma r^2 \epsilon$ in every iteration of Algorithm 3.1. Let*

---

[2]Strictly speaking, strong convexity is not required for Proposition 1. That is, the far left term in (3.6) can be replaced with zero.

v_p be the optimal value of Program (1.1). Let also v_{CGM}^l be the optimal value of Program (3.3). Then, at iteration $l \geq 1$, it holds that

(3.8)
$$v_{CGM}^l - v_p \leq \frac{4\gamma r^2 (1+\epsilon)}{l+2}.$$

Assuming that L satisfies (3.6), it is not difficult to verify that Program (1.1) is a convex and strongly smooth problem. Therefore CGM achieves the same convergence rate of $1/l$ that the projected gradient descent achieves for such problems [34]. We note that, under stronger assumptions, CGM can achieves linear convergence rate [35, 36].

A benefit of directly working with the infinite-dimensional Program (1.1) is that it provides a unified framework for various finite-dimensional approximations, such as the moments method [6]. In the context of CGM, following our discussion at the end of Section 2, a common approach to solve Program (3.2) is to search for an $O(\epsilon)$-approximate global solution over a finite grid on $\mathbb{I}$, as indicated in Proposition 1. The tractability of this gridding approach largely depends on how smooth $\Phi(t)$ is as a function of t, measured by its Lipschitz constant, which we denote by ϕ. Roughly speaking, to find an $O(\epsilon)$-approximate global solution of Program (3.2), one needs to search over a uniform grid of size $O(\phi/\epsilon)^{\dim(\mathbb{I})}$. As the dimension grows, the Lipsichtz constant ϕ must be smaller and smaller for this brute force search to be tractable. In some important applications, the dimension $\dim(\mathbb{I})$ is in fact small. In radar, array signal processing, or imaging applications, for example, $\dim(\mathbb{I}) \leq 2$.

As a numerical example with $\dim(\mathbb{I}) = 1$, let us revisit the setup in Section 2 with the choice of

$$\dot{x} = \frac{1}{4}(\delta_{0.1\pi} + \delta_{0.2\pi} + \delta_{0.3\pi} + \delta_{0.31\pi}),$$

$$\Phi(t) = [\ e^{-\pi\,\mathrm{i}(m-1)t} \quad \cdots \quad e^{\pi\,\mathrm{i}(m-1)t}\ ]^\top \in \mathbb{C}^m,$$

where $m = 33$. This Φ might be considered as a generic model for a sensing device and the resulting loss of low-frequency details [6]. Here, $\top$ stands for vector transpose. We solve Program (1.1) by applying Algorithm 3.1, where Program (3.2) therein is solved on uniform grids with sizes $\{10^2, 10^3, 10^4\}$. The recovery error in 1-Wasserstein metric, namely, $d_W(x^l, \dot{x})$, is shown in Figure 4.1a. The same experiment is repeated in Figure 4.1b after adding additive white Gaussian noise with variance of 0.01 to each coordinate of $\dot{y}$, see (2.3). Not surprisingly, the gains obtained from finer grids are somewhat diminished by the large noise. Both experiments were performed on a MacBook Pro (15-inch, 2017) with standard configurations. Section 8 outlines a few ideas for incorporating the continuous nature of $\mathbb{I}$ to develop new variants of CGM that would replace the naive gridding approach above.

4. Exchange Method. EM is a well-known algorithm to solve SIPs and, in particular, Program (1.3). In every iteration, EM adds a new constraint out of the infinitely many in Program (1.3), thereby forming an increasingly finer discretisation of $\mathbb{I}$ as the algorithm proceeds. The new constraints are added where needed most, namely, at $t \in \mathbb{I}$ that maximally violates the constraints in Program (1.3). In other words, a new constraint is added at $t \in \mathbb{I}$ that maximizes $\mathrm{Re}\langle \lambda^l, \Phi(t)\rangle$, where (λ^l, α^l) is the current iterate. EM is summarized in Algorithm 4.1.

Let (λ_d, α_d) be a maximizer of Program (1.3). Also assume that $T_d \subset \mathbb{I}$ is the set of *active* constraints in Program (1.3), namely $\mathrm{Re}\langle \lambda_d, \Phi(t)\rangle = \alpha_d$ for every $t \in T_d$. If an oracle tells us what the active constraints T_d are in advance, we can simply find the optimal pair (λ_d, α_d) by solving Program (1.3) with T_d instead of $\mathbb{I}$. Alas, such an oracle is not at hand. Instead, at iteration l, Algorithm 4.1

1. solves Program (1.3) restricted to the current constraints T^{l-1} to find (λ^l, α^l), and then
2. if (λ^l, α^l) does not violate the constraints of Program (1.3) on $\mathbb{I}\backslash T^{l-1}$, the algorithm terminates because it has found a maximizer of Program (1.3), namely (λ^l, α^l). Otherwise, EM adds to its support a point $t^l \in \mathbb{I}$ that maximally violates the constraints of Program (1.3).

(a) Noise-free

(b) Noisy

Figure 4.1: Recovery error in 1-Wasserstein metric using Algorithm 3.1 for the numerical example detailed at the end of Section 3. Grid sizes are given in the legends.

Having reviewed both CGM and EM for solving Program (1.1) in the past two sections, we next establish their equivalence.

5. Equivalence of CGM and EM. CGM solves Program (1.1) and adds a new atom in every iteration whereas EM solves the dual problem (namely Program (1.3)) and adds a new active constraint in every iteration, and both algorithms do so "greedily". Their connection goes deeper: Consider Program (1.1) restricted to a finite support $T \subset \mathbb{I}$, namely, the program

$$(5.1) \qquad \begin{cases} \min_{x} & L\left(\int_{\mathbb{I}} \Phi(t)x(dt) - y\right) \\ \text{subject to} & \|x\|_{TV} \leq 1 \\ & \text{supp}(x) \subseteq T \\ & x \in B_+(\mathbb{I}). \end{cases}$$

8

Algorithm 4.1 EM for solving Program (1.3).

Input: Compact set $\mathbb{I}$, continuous functions $\Phi : \mathbb{I} \to \mathbb{C}^m$ and $w : \mathbb{I} \to \mathbb{R}_{++}$, differentiable function $L : \mathbb{C}^m \to \mathbb{R}$, $y \in \mathbb{C}^m$ and tolerance $\eta \geq 0$.

Output: Vector $\widehat{\lambda} \in \mathbb{C}^m$ and $\widehat{\alpha} \geq 0$.

Initialize: $l = 1$ and $T^0 = \emptyset$.

While $\max\limits_{t \in \mathbb{I}} \mathrm{Re} \left\langle \lambda^l, \Phi(t) \right\rangle > \alpha^l + \eta$, **do**
  1. Let (λ^l, α^l) be a maximizer of

$$
(4.1) \qquad \begin{cases} \max\limits_{\lambda, \alpha} & \mathrm{Re} \left\langle \lambda, y \right\rangle - L_\circ(-\lambda) - \alpha \\ \text{subject to} & \mathrm{Re} \left\langle \lambda, \Phi(t) \right\rangle \leq \alpha \qquad t \in T^{l-1} \\ & \alpha \geq 0, \end{cases}
$$

  where $L_\circ$ is the Fenchel conjugate of L, see (1.4).
  2. Let t^l be the solution to

$$
(4.2) \qquad\qquad \max\limits_{t \in \mathbb{I}} \mathrm{Re} \left\langle \lambda^l, \Phi(t) \right\rangle .
$$

  3. Set $T^l = T^{l-1} \cup t^l$.
Return: $(\widehat{\lambda}, \widehat{\alpha}) = (\lambda^l, \alpha^l)$.

The dual of Program (5.1) is

$$
(5.2) \qquad \begin{cases} \max\limits_{\lambda, \alpha} & \mathrm{Re} \left\langle \lambda, y \right\rangle - L_\circ(-\lambda) - \alpha \\ \text{subject to} & \mathrm{Re} \left\langle \lambda, \Phi(t) \right\rangle \leq \alpha \qquad t \in T \\ & \alpha \geq 0. \end{cases}
$$

Indeed, Program (5.2) is the restriction of Program (1.3) to T. Note that the complementary slackness forces any minimizer of Program (5.1) to be supported on the set of active constraints of Program (5.2). Note also that Programs (5.1) and (5.2) appear respectively in CGM and EM but with different support sets. The following result states that CGM and EM are in fact equivalent algorithms to solve Program (1.1), see Appendix C for the proof.

PROPOSITION 2. **(Equivalence of Algorithms 3.1 and 4.1)** *For $\gamma \geq 1$, suppose that L satisfies (3.6). Assume also that CGM and EM update their supports according to the same rule, e.g., selecting the smallest solutions if $\mathbb{I} \subset \mathbb{R}$. Then CGM and EM are equivalent in the sense that $T^l_{CGM} = T^l_{EM}$ for every iteration $l \geq 0$. Here, T^l_{CGM} and T^l_{EM} (both subsets of $\mathbb{I}$) are the support sets of CGM and EM at iteration l, respectively.*

Furthermore, $v^l_{CGM} = v^{l+1}_{EM}$, where v^l_{CGM} and v^l_{EM} denote the optimal values of Programs (3.3) and (4.1) in CGM and EM, respectively.

The above equivalence allows us to carry convergence results from one algorithm to another. In particular, the convergence rate of CGM in Proposition 1 determines the convergence rate of EM, as the following result indicates, see Appendix D for the proof.

PROPOSITION 3. **(Convergence of Algorithm 4.1)** *For $\gamma \geq 1$, suppose that L satisfies (3.6). Recall the definition of r in (3.7) and, for $\epsilon \geq 0$, suppose that Program (4.2) is solved to within an accuracy of $2\gamma r^2 \epsilon$ in every iteration. Let v_d be the optimal value of Program (1.3) and (λ_d, α_d) be its unique maximizer. Likewise, let v^l_{EM} be the optimal value of Program (4.1). At iteration $l \geq 1$, it then holds that*

$$
(5.3) \qquad\qquad v^l_{EM} - v_d \leq \frac{4\gamma r^2(1+\epsilon)}{l+2},
$$

$$\|\lambda^l - \lambda_d\|_2 \leq \sqrt{\frac{8\gamma^2 r^2(1+\epsilon)}{l+2}},$$

(5.4)
$$|\alpha^l - \alpha_d| \leq \sqrt{\frac{8\gamma^2 r^4(1+\epsilon)}{l+2}}.$$

Furthermore, it holds that

(5.5)
$$\max_{t \in \mathbb{I}} \langle \lambda^l, \Phi(t) \rangle \leq \alpha_d + \sqrt{\frac{8\gamma^2 r^4(1+\epsilon)}{l+2}}.$$

That is, the iterates $\{\lambda^l\}_l$ of Algorithm 4.1 gradually become feasible for Program (1.3).

Proposition 3 states that Program (1.3), which has infinitely many constraints, can be solved as fast as a smooth convex program with finitely many constraints. More specifically, it is not difficult to verify that the objective function of Program (1.3) is convex and strongly smooth, see Section 7. Then, (5.3) states that EM solves Program (1.3) at the rate of $1/l$, the same rate at which the projected gradient descent solves a finite-dimensional problem under the assumptions of convexity and strong smoothness [34]. This is perhaps remarkable given that Program (1.3) has infinitely many constraints. Note however that the convergence of the iterates $\{(\lambda^l, \alpha^l)\}_l$ of EM to the unique maximizer (λ_d, α_d) of Program (1.3) is much slower as given in (5.4), namely, at the rate of $1/\sqrt{l}$.

We remark that Proposition 3 is novel in providing a rate of convergence for EM for a general class of nonlinear SIPs, whereas the literature on SIP only gives rates of convergence for specific problems. See Section 6 for a more detailed literature review.

6. Related Work. The conditional gradient method (CGM), also known as the Frank-Wolfe algorithm, is one of the earliest algorithms for constrained optimization [10]. The version of the algorithm considered in this paper is the fully-corrective Frank-Wolfe algorithm, also known as the *simplicial decomposition* algorithm, in which the objective is optimized over the convex hull of all previous atoms [11, 37]. The algorithm was proposed for optimization over measures, the context considered in this paper, in [4].

Semi-infinite programs (SIPs) have been much studied, both theoretically in terms of optimality conditions and duality theory, and algorithmically in terms of design and analysis of numerical methods for their solution. We refer the reader to the review articles [12] and [13] for further background.

Exchange methods are one of the three families of popular methods for the numerical solution of SIPs, with the other two being discretisation methods and localization methods. In discretisation methods, the infinitely many constraints are replaced by a finite subset thereof and the resulting finite dimensional problem is solved as an approximation of the SIP. In localization methods, a sequence of local (usually quadratic) approximations to the problem are solved.

Global convergence of discretisation methods has been proved for linear SIPs [38], but no general convergence result exists for nonlinear SIPs [12]. Global convergence of exchange methods has been proved for general SIPs [12], but to the authors' best knowledge there is no general proof of rate of convergence, except for more specific problems. For localization methods, local superlinear convergence has been proved assuming strong sufficient second-order optimality conditions, which do not hold for all SIPs [39]. The guarantees extend to global convergence of more sophisticated algorithms which combine localization methods with global search, see [13, Section 7.3] and references therein. We refer the reader to [12, 13] for more details on existing convergence analysis of SIPs. Against this background, the convergence rate of the EM, established here in Proposition 3 for a wide class of nonlinear SIPs, represents a new contribution.

The exchange method described in this paper can also be viewed as the cutting plane method, also known as Kelley's method [40, 41] applied to Program (1.3). Dual equivalence of conditional gradient methods and cutting plane methods is well known for finite-dimensional problems, see for example [42, 37, 43], and these results agree with the dual equivalence established in this paper.

EM may also be viewed as a bundle method for unconstrained optimization [44, 45]. Bundle methods construct piecewise linear approximations to an objective function using a "bundle" of subgradients from previous iterations. As a special case, given a convex and smooth function u and convex (but not necessarily

smooth) function v, the function $u + v$ may be minimized by constructing piecewise linear approximations to v, generating the sequence of iterates $\{\lambda^l\}_l$ specified as

$$(6.1) \qquad \lambda^l \in \arg\min_{\lambda} \left(u(\lambda) + \max_{1 \leq i \leq l-1} \mathrm{Re}\langle \lambda, \partial v(\lambda^i) \rangle \right),$$

where $\partial v(\lambda^i)$ is a subgradient of v at λ^i. To establish the connection with EM, note that Program (1.3) can be rewritten as the unconstrained problem

$$(6.2) \qquad \max_{\lambda \in \mathbb{C}^m} \ \mathrm{Re}\langle \lambda, y \rangle - L_\circ(-\lambda) - \max_{t \in \mathbb{I}}\langle \lambda, \Phi(t) \rangle.$$

Setting $u(\lambda) = -\mathrm{Re}\langle \lambda, y \rangle + L_\circ(-\lambda)$ and $v(\lambda) = \max_{t \in \mathbb{I}}\langle \lambda, \Phi(t) \rangle$, and then applying the bundle method produces the iterates

$$(6.3) \qquad \lambda^l \in \arg\max \mathrm{Re}\langle \lambda, y \rangle - L_\circ(-\lambda) - \max_{t \in T^l}\langle \lambda, \Phi(t) \rangle \equiv \text{Program (4.1)}.$$

That is, EM applied to Program (1.3) and the bundle method described above applied to Program (6.2) produce the same iterates. The dual equivalence of CGM and the bundle method has previously been noted for various finite dimensional problems, see for example [45, Chapter 7]. However, we are not aware of any extension of these finite-dimensional results to SIPs and their dual problem of optimization over Borel measures. In this sense, the equivalence established in Proposition 2 is novel.

7. Geometric Insights. This section collects a number of useful insights about CGM/EM and Program (1.1) in the special case where $\mathbb{I} \subset \mathbb{R}$ is a compact subset of the real line and the function $\Phi : \mathbb{I} \to \mathbb{C}^m$ is a *Chebyshev system* [46], see Section 1.

DEFINITION 7.1. **(Chebyshev system)** *Consider a compact interval $\mathbb{I} \subset \mathbb{R}$ and a continuous function $\Phi : \mathbb{I} \to \mathbb{C}^m$. Then Φ is a Chebyshev system if $\{\Phi(t_i)\}_{i=1}^m \subset \mathbb{C}^m$ are linearly independent vectors for any choice of distinct $\{t_i\}_{i=1}^m \subset \mathbb{I}$.*[3]

Chebyshev systems are widely used in classical approximation theory and generalize the notion of ordinary polynomials. Many functions form Chebyshev systems, for example sinusoids or translated copies of the Gaussian window, and we refer the interested reader to [46, 47, 1] for more on their properties and applications. Let $C_{\mathbb{I}} \subset \mathbb{C}^m$ be the convex hull of $\{\Phi(t)\}_{t \in \mathbb{I}} \cup \{0\}$, namely

$$(7.1) \qquad C_{\mathbb{I}} := \left\{ \int_{\mathbb{I}} \Phi(t)x(dt) : x \in B_+(\mathbb{I}), \|x\|_{TV} \leq 1 \right\}.$$

Note that $\{x \in B_+(\mathbb{I}) : \|x\|_{TV} \leq 1\}$ is a compact set. Then, by the continuity of Φ and with an application of the dominated convergence theorem, it follows that $C_{\mathbb{I}}$ is a compact set too. Since Φ is by assumption a Chebyshev system, $C_{\mathbb{I}} \subset \mathbb{C}^m$ is in fact a *convex body*, namely a compact convex set with non-empty interior. Introducing $z = \int_{\mathbb{I}} \Phi(t)x(dt)$, we note that Program (1.1) is equivalent to the program

$$(7.2) \qquad \min_{z \in C_{\mathbb{I}}} L(z - y).$$

The compactness of $C_{\mathbb{I}}$ and the strong convexity of L in (3.6) together imply that Program (7.2) has a unique minimizer $y_p \in C_{\mathbb{I}}$, which can be written as $y_p = \int_{\mathbb{I}} \Phi(t)x_p(dt)$, where x_p itself is a minimizer of Program (1.1). For example, when $L(\cdot) = \frac{1}{2}\| \cdot \|_2^2$, Program (7.2) projects y onto $C_{\mathbb{I}}$. That is, y_p is the orthogonal projection of y onto the convex set $C_{\mathbb{I}}$.

Given the equivalence of Programs (1.1) and (7.2), we might say that solving Program (1.1) "denoises" the signal y from a signal processing viewpoint, in the sense that it finds a nearby signal $y_p = \int_{\mathbb{I}} \Phi(t)x_p(dt)$ that has a sparse representation in the dictionary $\{\Phi(t)\}_{t \in \mathbb{I}}$ (because x_p is a sparse measure). To be more specific, by Carathéodory's theorem [48], every $y_p \in C_{\mathbb{I}}$ can be written as a convex combination of at most m atoms of the dictionary $\{\Phi(t)\}_{t \in \mathbb{I}}$. On the other hand, the Chebyshev assumption on Φ implies that $\{\Phi(t)\}_{t \in \mathbb{I}}$ are the *extreme points* of $C_{\mathbb{I}}$ [46, Chapter II]. Here, an extreme point of $C_{\mathbb{I}}$ is a point in $C_{\mathbb{I}}$ that cannot be

---

[3]Note that Definition 7.1 is slightly different from the standard one in [46] which requires Φ to be real-valued.

written as a convex combination of other points in $C_\mathbb{I}$. It then follows that this atomic decomposition of y_p is unique, and x_p is necessarily m-sparse. We may note the analogous result in the finite-dimensional case. Indeed, the Lasso problem is known to have a unique solution whose sparsity is no greater than the rank of the measurement matrix, provided the columns of the measurement matrix are in general position [49].

At iteration l of CGM, let $C^l \subset \mathbb{C}^m$ be the convex hull of $\{\Phi(t)\}_{t \in T^l} \cup \{0\}$, namely

$$(7.3) \qquad C^l := \left\{ \sum_{t \in T^l} \Phi(t) \cdot a_t : \sum_{t \in T^l} a_t \le 1, \, a_t \ge 0, \, \forall t \in T^l \right\}.$$

Similar to the argument above, we observe that Program (3.3) is equivalent to

$$(7.4) \qquad \min_{z \in C^l} L(z - y).$$

As with Program (7.2), Program (7.4) has a unique minimizer $y^l \in C^l$ that satisfies $y^l = \int_{T^l} \Phi(t) x^l(dt)$ and x^l is a minimizer of Program (3.3). By Carathéodory's theorem again, x^l is at most m-sparse. In other words, there always exists an m-sparse minimizer x^l to Program (3.3); iterates of CGM are always sparse and so are the iterates of EM by their equivalence in Proposition 2.

In addition, note that the chain $C^1 \subseteq C^2 \subseteq \cdots \subseteq C_\mathbb{I}$ provides a sequence of increasingly better approximations to $C_\mathbb{I}$. CGM eventually terminates when $y_p = y^l \in C^l \subseteq C_\mathbb{I}$, which happens as soon as C^l contains the *face* of $C_\mathbb{I}$ to which y^p belongs. It is however common to use different stopping criteria to terminate CGM when y^l is sufficiently close to y_p.

Let us now rewrite Program (1.3) in a similar way. First let $C_{\mathbb{I},\circ} \subset \mathbb{C}^m$ be the *polar* of $C_\mathbb{I}$, namely

$$C_{\mathbb{I},\circ} = \{\lambda : \operatorname{Re} \langle \lambda, z \rangle \le 1, \, \forall z \in C_\mathbb{I}\} = \{\lambda : \operatorname{Re} \langle \lambda, \Phi(t) \rangle \le 1, \, \forall t \in \mathbb{I}\},$$

where the second identity follows from the definition of $C_\mathbb{I}$. Let also $g_{C_{\mathbb{I},\circ}} = \gamma_{C_\mathbb{I}}$ denote the *gauge function* associated with $C_{\mathbb{I},\circ}$ and the *support function* associated with $C_\mathbb{I}$, respectively [50]. That is, for $\lambda \in \mathbb{C}^m$, we define

$$(7.5) \qquad g_{C_{\mathbb{I},\circ}}(\lambda) := \begin{cases} \min_{\alpha} & \alpha \\ \text{subject to} & \lambda \in \alpha \cdot C_{\mathbb{I},\circ} \\ & \alpha \ge 0 \end{cases} = \max_{t \in \mathbb{I}} \operatorname{Re}\langle \lambda, \Phi(t) \rangle = \max_{z \in C_\mathbb{I}} \operatorname{Re}\langle \lambda, z \rangle =: \gamma_{C_\mathbb{I}}(\lambda),$$

where $\alpha \cdot C_{\mathbb{I},\circ} = \{\alpha\lambda : \lambda \in C_{\mathbb{I},\circ}\}$. In words, $g_{C_{\mathbb{I},o}}(\lambda) = \gamma_{C_\mathbb{I}}(\lambda)$ is the smallest α at which the inflated "ball" $\alpha \cdot C_{\mathbb{I},\circ}$ first reaches λ. By usual convention, the optimal value above is set to infinity when the problem is infeasible, namely when the ray that passes through λ does not intersect $C_{\mathbb{I},\circ}$. It is also not difficult to verify that $g_{C_{\mathbb{I},\circ}} = \gamma_{C_\mathbb{I}}$ is a positively-homogeneous convex function. Using (7.5), we may rewrite Program (1.3) as

$$(7.6) \qquad \begin{cases} \max_{\lambda,\alpha} & L_\circ(-\lambda) + \operatorname{Re}\langle \lambda, y \rangle - \alpha \\ \text{subject to} & \lambda \in \alpha \cdot C_{\mathbb{I},\circ} \\ & \alpha \ge 0 \end{cases} \equiv \max_{\lambda \in \mathbb{C}^m} \operatorname{Re}\langle \lambda, y \rangle - L_\circ(-\lambda) - g_{C_{\mathbb{I},\circ}}(\lambda).$$

By the assumption in (3.6), L is strongly smooth and consequenly $L_\circ$ is strongly convex [34]. Therefore Program (1.3) has a unique maximizer, which we denote by (λ_d, α_d). The optimality of (λ_d, α_d) also immediately implies that

$$(7.7) \qquad \alpha_d = g_{C_{\mathbb{I},\circ}}(\lambda_d).$$

Thanks to Proposition 2, we likewise define the polar of C^{l-1} and the corresponding gauge function to rewrite the main step of EM in Algorithm 4.1, namely

$$(7.8) \qquad \begin{cases} \max_{\lambda,\alpha} & L_\circ(-\lambda) + \operatorname{Re}\langle \lambda, y \rangle - \alpha \\ \text{subject to} & \lambda \in \alpha \cdot C_\circ^{l-1} \\ & \lambda \ge 0 \end{cases} \equiv \max_{\lambda \in \mathbb{C}^m} \operatorname{Re}\langle \lambda, y \rangle - L_\circ(-\lambda) - g_{C_\circ^{l-1}}(\lambda)$$

and the unique minimizer of the above three programs is (λ^l, α^l), where the uniqueness again comes from the strong convexity of $L_\circ$. Similar to (7.7), the optimality of (λ^l, α^l) immediately implies that

$$(7.9) \qquad\qquad \alpha^l = g_{C_\circ^l}(\lambda^l).$$

It is not difficult to verify that

$$(7.10) \qquad\qquad C_\circ^1 \supseteq C_\circ^2 \supseteq \cdots \supseteq C_{\mathbb{I},\circ}.$$

That is, as EM progresses, C^l gradually "zooms into" $C_{\mathbb{I},\circ}$. As with CGM, EM eventually terminates as soon as $C_\circ^l$ includes the face of $C_{\mathbb{I},\circ}$ to which λ_d/α_d belongs, at which point $(\lambda^l, \alpha^l) = (\lambda_d, \alpha_d)$. In light of the argument in Appendix C, in every iteration, we also have that

$$(7.11) \qquad\qquad \langle y^l, \lambda^l/\alpha^l \rangle = 1,$$

namely the pair $(y^l, \lambda^l/\alpha^l) \in C^l \times C_\circ^l$ satisfies the generalized Holder inequality $g_{C^l} \cdot g_{C_\circ^l} \le 1$ with equality [50]. Here, g_{C^l} and $g_{C_\circ^l}$ are the gauge functions of C^l and $C_\circ^l$, respectively. It is worth pointing out that, with the choice of $L(\cdot) = L_\circ(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, the maximizer of (7.8) is the same as the (unique) minimizer of

$$\min_{\lambda \in \mathbb{C}^m} \frac{\|\lambda - y\|_2^2}{2} + g_{C_\circ^{l-1}}(\lambda),$$

which might be interpreted as a generalization of Lasso and other standard tools for sparse denoising [19]. That is, each iteration of CGM/EM can be interpreted as a simple denoising procedure.

8. Future Directions. Even though, by Proposition 1, CGM reduces the objective function of Program (1.1) at the rate of $O(1/l)$, the behavior of $\{x^l\}_l$, namely, the sequence of measures generated by Algorithm 3.1, is often less than satisfactory. Indeed, in practice, the greedy nature of CGM leads to adding clusters of spikes to the support, many of which are spurious.

In this sense, all applications reviewed in Section 2 will benefit from improving the performance of CGM. In particular, a variant suggested in [4] follows each step of CGM with a heuristic local search. Intuitively, this modification makes the algorithm less greedy and helps avoid the clustering of spikes, described above.

The equivalence of CGM and EM discussed in this paper might offer a new and perhaps less heuristic approach to improving CGM. From the perspective of EM, a natural improvement to Algorithm 4.1 (and consequently Algorithm 3.1) might be obtained by replacing Program (4.1) with

$$(8.1) \qquad \begin{cases} \max\limits_{\lambda, \alpha} & \operatorname{Re} \langle \lambda, y \rangle - L_\circ(-\lambda) - \alpha \\ \text{subject to} & \operatorname{Re} \langle \lambda, \Phi(t) \rangle \le \alpha \qquad t \in T_\delta^{l-1} \\ & \alpha \ge 0, \end{cases}$$

where $T_\delta^{l-1} \subseteq \mathbb{I}$ is the δ-neighborhood of the current support T^{l-1}, namely, all the points in $\mathbb{I}$ that are within δ distance of the set T^{l-1}.

At the first glance, Program (8.1) is itself a semi-infinite program and not any easier to solve than Program (1.3). However, if δ is sufficiently small compared to the Lipschitz constant of Φ, then one might use a local approximation for Φ to approximate Program (8.1) with a finite-dimensional problem. For instance, if Φ is differentiable, one could use the first order Taylor expansion of Φ around each impulse in T^{l-1}. As another example, suppose that $\Phi : \mathbb{I} = [0,1) \to \mathbb{C}^m$ and specified as

$$(8.2) \qquad \Phi(t) = [\; e^{-\pi \mathrm{i}(m-1)t} \quad \cdots \quad e^{\pi \mathrm{i}(m-1)t} \;]^\top \in \mathbb{C}^m,$$

see the numerical test at the end of Section 3. It is easy to verify that $\{\Phi(j/m)\}_{j=0}^{m-1}$ form an orthonormal basis for $\mathbb{C}^m$. Even though we may represent $\Phi(t)$ in Program (8.1) within this basis for any $t \in \mathbb{I}$, this representation is not "local" [29, 20]. A better local representation of $\Phi(t)$ within a δ-neighborhood is obtained through the machinery of *discrete prolate spheroidal wave functions* [51].

In light of this discussion, an interesting future research direction might be to study variants of Program (8.1) and their potential impact in various applications.

13

REFERENCES

[1] A. Eftekhari, J. Tanner, A. Thompson, B. Toader, and H. Tyagi. Sparse non-negative super-resolution — simplified and stabilised. *arXiv preprint arXiv:1804.01490*, 2018.

[2] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[3] G. Schiebinger, E. Robeva, and B. Recht. Superresolution without separation. *Information and Inference*, page iax006, 2017.

[4] N. Boyd, G. Schiebinger, and B. Recht. The alternating descent conditional gradient method for sparse inverse problems. *SIAM Journal on Optimization*, 27(2):616–639, 2017.

[5] E. Candès and C. Fernandez-Granda. Super-resolution from noisy data. *Journal of Fourier Analysis and Applications*, 19(6):1229–1254, 2013.

[6] E. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6):906–956, 2014.

[7] Y. De Castro and F. Gamboa. Exact reconstruction using Beurling minimal extrapolation. *Journal of Mathematical Analysis and applications*, 395(1):336–354, 2012.

[8] C. Fernandez-Granda. Support detection in super-resolution. In *Proceedings of the 10th International Conference on Sampling Theory and Applications*, 2013.

[9] Q. Denoyelle, V. Duval, and G. Peyré. Support recovery for sparse deconvolution of positive measures. *Journal of Fourier Analysis and its Applications*, 23(5):1153–1194, 2017.

[10] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[11] C. Holloway. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6(1):14–27, 1974.

[12] R. Hettich and K. Kortanek. Semi-infinite programming: theory, methods and applications. *SIAM Review*, 35(3):380–429, 1993.

[13] M. Lopez and G. Still. Semi-infinite programming. *European Journal of Operational Research*, 180(2):491–518, 2005.

[14] Christodoulos A Floudas and Oliver Stein. The adaptive convexification algorithm: a feasible point method for semi-infinite programming. *SIAM Journal on Optimization*, 18(4):1187–1208, 2007.

[15] V. Duval. A characterization of the non-degenerate source condition in super-resolution. *arXiv preprint arXiv:1712.06373*, 2017.

[16] E.J. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6):906–956, 2014.

[17] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[18] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier Science, 2008.

[19] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2015.

[20] Z. Zhu and M. Wakin. Approximating sampled sinusoids and multiband signals using multiband modulated DPSS dictionaries. *Journal of Fourier Analysis and Applications*, 23(6):1263–1310, 2017.

[21] M. Slawski and M. Hein. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

[22] Emmanuel J Candes, Yonina C Eldar, Deanna Needell, and Paige Randall. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, 31(1):59–73, 2011.

[23] Gongguo Tang, Badri Narayan Bhaskar, and Benjamin Recht. Sparse recovery over continuous dictionaries-just discretize. In *Signals, Systems and Computers, 2013 Asilomar Conference on*, pages 1043–1047. IEEE, 2013.

[24] G. Tang, B.N. Bhaskar, P. Shah, and B. Recht. Compressed sensing off the grid. *IEEE Transactions on Information Theory*, 59(11):7465–7490, 2013.

[25] V. Duval and G. Peyré. Sparse spikes super-resolution on thin grids II: the continuous basis pursuit. *Inverse Problems*, 33(9):095008, 2017.

[26] V. Duval and G. Peyré. Sparse spikes super-resolution on thin grids I: the lasso. *Inverse Problems*, 33(5):055008, 2017.

[27] A. Eftekhari, J. Romberg, and M.B. Wakin. Matched filtering from limited frequency samples. *IEEE Transactions on Information Theory*, 59(6):3475–3496, 2013.

[28] A. Eftekhari and M.B. Wakin. Supplementary material for "Greed is super: A fast algorithm for super-resolution". Technical report, Colorado School of Mines, 2015.

[29] A. Eftekhari and M.B. Wakin. Greed is super: A new iterative method for super-resolution. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[30] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[31] Kristian Bredies and Hanna Katriina Pikkarainen. Inverse problems in spaces of measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(1):190–218, 2013.

[32] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions*

on information theory, 53(12):4655–4666, 2007.

[33] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[34] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2013.

[35] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. *arXiv preprint arXiv:1406.1305*, 2014.

[36] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[37] D. Bertsekas and H. Yu. A unifying polyhedral approximation framework for convex optimization. *SIAM Journal on Optimization*, 21(1):333–360, 2011.

[38] M. Goberna and M. Lopez. *Linear semi-infinite optimization*. Wiley, Chichester, 1998.

[39] R. Fontecilla, T. Steihaug, and R. Tapia. A convergence theory for a class of quasi-Newton methods for constrained optimization. *SIAM Journal on Numerical Analysis*, 24(5):1133–1151, 1987.

[40] J. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society of Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[41] C.T. Kelley. *Iterative methods for optimization*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1999.

[42] F. Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.

[43] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *international Conference on Machine Learning*, 2013.

[44] A. Bagirov, N. Karmitsa, and M.M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. SpringerLink : Bücher. Springer International Publishing, 2014.

[45] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

[46] S. Karlin and W. Studden. *Tchebycheff systems: with applications in analysis and statistics*. Pure and applied mathematics. Interscience Publishers, 1966.

[47] M. Krein and A. Nudelman. *The Markov moment problem and extremal problems*. American Mathematical Society, 1977.

[48] A. Barvinok. *A Course in Convexity*. Graduate studies in mathematics. American Mathematical Society, 2002.

[49] R. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7, 2012.

[50] R. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 2015.

[51] D. Slepian. Prolate spheroidal wave functions, Fourier analysis and uncertainty V. *Bell Systems Technical Journal*, 57(5):1371–1429, 1978.

Appendix A. Proof of Proposition 1.

Recall the equivalent form of Program (1.1) given by Program (7.2), and let v_p be the optimal value of both these programs. We first establish that the iterates x^i of CGM applied to (1.1) are related to the iterates z^i of CGM applied to (3.4) by $z^i = \int_{\mathbb{I}} \Phi(t) x^i(dt)$. In this regard, suppose that $z^i = \int_{\mathbb{I}} \Phi(t) x^i(dt)$ and let t^{i+1} be the solution to Program (3.2). Let s^i be the output of the linear minimization step of CGM applied to Program (7.2). Then

$$s^i = \arg\min_{s \in \mathbb{C}_{\mathbb{I}}} \langle s, \nabla L(z^i - y) \rangle = \Phi(t^{i+1}),$$

which shows that the linear steps of CGM for both formulations coincide.

Now suppose that Program (3.2) is solved to an accuracy of $\theta \cdot \epsilon$ in every iteration, where

$$(A.1) \qquad \theta = \begin{cases} \sup_{\rho, z, s} & \frac{2}{\rho^2} \left(L(z' - y) - L(z - y) - \langle z' - z, \nabla L(z - y) \rangle \right) \\ & z' = z + \rho(s - z) \\ & z, s \in C_{\mathbb{I}} \\ & \rho \in [0, 1]. \end{cases}$$

Then we may invoke Theorem 1 in [33] to obtain that

$$v^l_{CGM} - v_p \leq \frac{2\theta(1 + \epsilon)}{l + 2}.$$

Let us next bound θ in terms of the known quantities. Due to the assumption (3.6), for any feasible pair

(z, z') in (A.1), we have that

$$L(z' - y) - L(z - y) - \langle z' - z, \nabla L(z - y) \rangle \leq \frac{\gamma}{2} \|z' - z\|_2^2$$

$$\leq \frac{\gamma \rho^2}{2} \|s - z\|_2^2 \qquad (z' = z + \rho(s - z))$$

$$\leq \frac{\gamma}{2} (\|s\|_2^2 + \|z\|_2^2)$$

(A.2)
$$\leq \gamma \rho^2 r^2, \qquad (\text{see (A.1)})$$

which immediately implies that $\theta \leq 2\gamma r^2$, and (3.8) now follows.

Appendix B. Duality of Programs (1.1) and (1.3).

We show here that the dual of Program (1.1) is Program (1.3). We first observe that Program (1.1) is equivalent to

(B.1)
$$\begin{cases} \min_{z,x} & L(z - y) \\ \text{subject to} & z = \int_{\mathbb{I}} \Phi(t) x(dt) \\ & \|x\|_{TV} \leq 1 \\ & x \in B_+(\mathbb{I}). \end{cases}$$

Introducing Lagrange multipliers $\lambda \in \mathbb{C}^m$ and $\alpha \geq 0$ for the two respective constraints, the Lagrangian $\mathcal{L}(x, \lambda, \alpha)$ for Program (B.1) is

$$\mathcal{L}(x, \lambda, \alpha) = L(z - y) - \text{Re} \left\langle \lambda, \left(z - \int_{\mathbb{I}} \Phi(t) x(dt) \right) \right\rangle + \alpha \left(\|x\|_{TV} - 1 \right)$$

$$= L(z - y) + \text{Re}\langle \lambda, z \rangle + \int_{\mathbb{I}} (\alpha - \text{Re}\langle \lambda, \Phi(t) \rangle) \, x(dt),$$

and so the dual of Program (B.1) is

$$\max_{\lambda \in \mathbb{C}^m, \alpha \geq 0} \left\{ \inf_{z \in \mathbb{C}^m} [L(z - y) + \text{Re}\langle \lambda, z - y \rangle] + \inf_{\mu \in B_+(\mathbb{I})} \left[\int_{\mathbb{I}} (\alpha - \text{Re}\langle \lambda, \Phi(t) \rangle) \, x(dt) \right] + \text{Re}\langle \lambda, y \rangle - \alpha \right\}$$

where $B_+(\mathbb{I})$ is the set of all nonnegative Borel measures supported on $\mathbb{I}$. Using the definition of the Fenchel conjugate in (1.4), the above problem is equivalent to

$$\begin{cases} \max_{\lambda, \alpha} & -L_0(-\lambda) + \text{Re}\langle \lambda, y \rangle - \alpha \\ \text{subject to} & \text{Re}\langle \lambda, \Phi(t) \rangle \leq \alpha \qquad t \in \mathbb{I} \\ & \alpha \geq 0, \end{cases}$$

which is Program (1.3).

Appendix C. Proof of Proposition 2.

By construction, $T_{CGM}^0 = T_{EM}^0 = \emptyset$. Fix iteration $l \geq 1$ and assume that $T_{CGM}^{l-1} = T_{EM}^{l-1} = T^{l-1}$. We next show that $T_{CGM}^l = T_{EM}^l = T^l = T^{l-1} \cup \{t^l\}$, namely the two algorithms add the same point t^l to their support sets in iteration l. We opt for a geometric argument here that relies heavily on Section 7.

Recall that Program (3.3) is equivalent to Program (7.4). Recall also that $y^l = \int_{T^l} \Phi(t) x^l(dt)$ is the unique minimizer of Program (7.4), where x^l is a minimizer of Program (3.3). On the other hand, recall that Program (4.1) is equivalent to Program (7.8), and both programs have the unique minimizer (λ^l, α^l). Since Program (4.1) only has linear constraints, Slater's condition is met and there is no duality gap between Programs (7.4) and (7.8). Furthermore, the tuple $(y^l, \lambda^l, \alpha^l)$ satisfies the KKT conditions, namely

$$y^l \in C^{l-1}, \qquad \lambda^l \in \alpha^l \cdot C_\circ^{l-1}, \qquad \alpha^l \geq 0,$$

$$\lambda^l = -\nabla L(y^l - y), \qquad \langle y^l, \lambda^l \rangle = \alpha^l,$$

16

From the above expression for λ^l, it follows immediately that the same point is added to the support in both Programs (3.3) and (4.1), which implies that $T_{CGM}^l = T_{EM}^l = T^{l-1} \cup \{t^l\}$. Finally, the above argument reveals that $v_{CGM}^l = v_{EM}^{l+1}$, which completes the proof of Proposition 2.

Appendix D. Proof of Proposition 3.

Note that

$$
\begin{aligned}
v_{EM}^l - v_d = v_{CGM}^{l-1} - v_d \qquad &\text{(see Proposition 2)} \\
= v_{CGM}^{l-1} - v_p \qquad &\text{(strong duality between Programs (1.1) and (1.3))} \\
\leq \frac{4\gamma r^2(1+\epsilon)}{l+2}, \qquad &\text{(see Proposition 1)}
\end{aligned}
$$
(D.1)

which proves the first claim in Proposition 3. To prove the second claim there, first recall the setup in Section 7. Let us first show that the minimizer of Program (4.1), namely (λ^l, α^l), converge to the minimizer of Program (1.3), namely (λ_d, α_d). To that end, recall the equivalent formulation of Programs (1.3,4.1) given in (7.6,7.8), and let

$$
h_{C_{\mathbb{I},o}}(\lambda) := \mathrm{Re}\langle \lambda, y\rangle - L_\circ(-\lambda) - g_{C_{\mathbb{I},o}}(\lambda),
$$

$$
h_{C_\circ^l}(\lambda) := \mathrm{Re}\langle \lambda, y\rangle - L_\circ(-\lambda) - g_{C_\circ^{l-1}}(\lambda),
$$
(D.2)

denote their objective functions, respectively. In particular, note that

$$
h_{C_{\mathbb{I},o}}(\lambda_d) = v_d, \qquad h_{C_\circ^l}(\lambda^l) = v_{EM}^l.
$$
(D.3)

By assumption in (3.6), L is γ-strongly smooth and therefore $L_\circ$ is (γ^{-1})-strongly convex [34]. Consequently, $-h_{C_{\mathbb{I},o}}$ is also (γ^{-1})-strongly convex, which in turn implies that

$$
\begin{aligned}
\frac{1}{2\gamma}\|\lambda^l - \lambda_d\|_2^2 &\leq -h_{C_\circ^l}(\lambda_d) + h_{C_\circ^l}(\lambda^l) + \langle \lambda_d - \lambda^l, \nabla h_{C_\circ^l}(\lambda^l)\rangle \\
&= -h_{C_\circ^l}(\lambda_d) + v_{EM}^l, \qquad \text{(see (D.3))}
\end{aligned}
$$
(D.4)

where the inner product above disappears by optimality of λ^l in Program (7.8). Let us next control $h_{C_\circ^l}(\lambda_d)$ in the last line above by noting that

$$
\begin{aligned}
h_{C_\circ^l}(\lambda^d) = \mathrm{Re}\langle \lambda_d, y\rangle - L_\circ(-\lambda_d) - g_{C_\circ^l}(\lambda_d) \qquad &\text{(see (D.2))} \\
\geq \mathrm{Re}\langle \lambda_d, y\rangle - L_\circ(-\lambda_d) - g_{C_{\mathbb{I},o}}(\lambda_d) \qquad &\left(C_\circ^l \supseteq C_{\mathbb{I},o} \text{ in (7.10)}\right) \\
= h_{C_{\mathbb{I},o}}(\lambda_d) & \\
= v_d. \qquad &\text{(see (D.3))}
\end{aligned}
$$
(D.5)

By substituting the bound above back into (D.4), we find that

$$
\begin{aligned}
\|\lambda^l - \lambda_d\|_2^2 &\leq 2\gamma(v_{EM}^l - v_d) \\
&\leq \frac{8\gamma^2 r^2(1+\epsilon)}{l+2}. \qquad \text{(see (D.1))}
\end{aligned}
$$
(D.6)

The above bound also allows us to find the convergence rate of α^l to α_d. Indeed, note that

$$
\begin{aligned}
|\alpha^l - \alpha_d| = \left| g_{C_\circ^l}(\lambda^l) - g_{C_{\mathbb{I},o}}(\lambda_d) \right| \qquad &\text{(see (7.9,7.7))} \\
= \left| \max_{t \in T^l}\langle \lambda^l, \Phi(t)\rangle - \max_{t \in \mathbb{I}}\langle \lambda_d, \Phi(t)\rangle \right| \qquad &\text{(see (7.5))} \\
\leq \max_{t \in \mathbb{I}} \left| \langle \lambda^l - \lambda_d, \Phi(t)\rangle \right| & \\
\leq \|\lambda^l - \lambda_d\|_2 \max_{t \in \mathbb{I}} \|\Phi(t)\|_2 & \\
\leq \sqrt{\frac{8\gamma^2 r^2(1+\epsilon)}{l+2}} \cdot r. \qquad &\text{(see (D.6,3.7))}
\end{aligned}
$$
(D.7)

With an argument similar to (D.7), we also find that

$$(D.8) \qquad \left| \max_{t \in \mathbb{I}} \langle \lambda^l, \Phi(t) \rangle - \alpha_d \right| \leq \sqrt{\frac{8\gamma^2 r^2 (1 + \epsilon)}{l + 2}} \cdot r,$$

which completes the proof of Proposition 3.