Journal of Machine Learning Research 23 (2022) 1-42 Submitted 2/21; Revised 10/22; Published 11/22

Joint Continuous and Discrete Model Selection via Submodularity

Jonathan Bunton J.BUNTON@UCLA.EDU
Paulo Tabuada TABUADA@EE.UCLA.EDU
Department of Electrical & Computer Engineering
University of California, Los Angeles
420 Westwood Plaza, Box 951594
Los Angeles, CA 90095-1554, USA

Editor: Andreas Krause

Abstract

In model selection problems for machine learning, the desire for a well-performing model with meaningful structure is typically expressed through a regularized optimization problem. In many scenarios, however, the meaningful structure is specified in some discrete space, leading to difficult nonconvex optimization problems. In this paper, we connect the model selection problem with structure-promoting regularizers to submodular function minimization with continuous and discrete arguments. In particular, we leverage the theory of submodular functions to identify a class of these problems that can be solved exactly and efficiently with an agnostic combination of discrete and continuous optimization routines. We show how simple continuous or discrete constraints can also be handled for certain problem classes, and extend these ideas to a robust optimization framework. We also show how some problems outside of this class can be embedded into the class, further extending the class of problems our framework can accommodate. Finally, we numerically validate our theoretical results with several proof-of-concept examples with synthetic and real-world data, comparing against state-of-the-art algorithms.

Keywords: Submodularity, submodular function minimization, mixed continuous discrete optimization, convex optimization, sparsity

1. Introduction

In many machine learning tasks, we require a model that not only performs a specified task well, but also has some meaningful structure. Models with meaningful structure can, for example, be easier to understand and implement. The desire for both accuracy and meaningful structure is usually expressed in a regularized optimization problem:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}}\ f(\mathbf{x}) + \lambda g(\mathbf{x}). \tag{1}$$

In this problem, $\mathbf{x}$ is a choice of model parameters from a parameter space $\mathcal{X}$, $f : \mathcal{X} \to \mathbb{R}$ is a function that describes the misfit of the model with the selected parameters to the given task (e.g., empirical risk), $g : \mathcal{X} \to \mathbb{R}$ is a function that expresses the deviation of our selected model parameters from some desired structure, and $\lambda \in \mathbb{R}_{\geq 0}$ is a tradeoff parameter.

Problem (1) becomes difficult when the desired model structure is an inherently discrete property, but the model parameters are continuous values $\mathbf{x}$ from a continuum $\mathcal{X}$. A prime example of this issue arises in feature selection for sparse regression, where we seek a linear predictor $\mathbf{x}^* \in \mathcal{X} \subseteq \mathbb{R}^n$ such that:

$$\mathbf{x}^* \in \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} \ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{2}$$

for some $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, with $\|\mathbf{x}\|_2$ the standard Euclidean norm on $\mathbb{R}^m$, and $\|\mathbf{x}\|_0$ the ℓ_0 pseudo-norm that counts the number of nonzero entries in the predictor $\mathbf{x}$. The desired structure, in this case, is a sparse predictor $\mathbf{x} \in \mathcal{X}$. Sparsity, however, only depends on the combinatorial choice of zero entries in the model parameters $\mathbf{x}$, whereas the model also requires a choice of continuous values for $\mathbf{x} \in \mathcal{X}$.

Problems with this mixed dependence on both continuous and discrete properties of the model parameters such as (2) are notoriously difficult, and even NP-Hard in general (Rauhut, 2010). A typical workaround is to replace the function describing model structure, g in problem (1), with a continuous relaxation that is more amenable to optimization. One of the more celebrated instances of this approach is the relaxation of the ℓ_0 pseudo-norm in (2) to the convex ℓ_1 norm $\|\mathbf{x}\|_1$, which instead sums the absolute values of the vector $\mathbf{x}$. While this relaxation still encourages the intended structure, the minimizer for the relaxed problem does not necessarily correspond to the minimizer for the initially specified problem (Bach et al., 2012). Moreover, the well-known conditions for sparse recovery in regression problems, such as Restricted Isometry Properties (Candes and Tao, 2005), Null Space Properties (Rauhut, 2010), and Irrepresentability Conditions (Zhao and Yu, 2006), are not applicable to more general discrete functions g.

In contrast, in this work we identify conditions that allow us to directly solve the originally posed regularized model-fitting problem (1) exactly and efficiently. To derive our new conditions, we leverage submodularity, a property of functions that defines a boundary between easy and hard optimization problems. Our approach stands in stark contrast to existing methods, which either focus on submodularity in purely one domain (Bach, 2019) or relies on restricted isometry or strong convexity constants that are NP-Hard to compute (Elenberg et al., 2018; El Halabi and Jegelka, 2020).

Traditionally, submodularity is defined for functions on bounded discrete sets, where arbitrary function minimization is NP-Hard. When a function is submodular, however, it can be minimized exactly in polynomial time (Schrijver, 2003). The definition of submodularity extends to continuous functions as well, and recently the associated optimization guarantees have also been extended (Bach, 2019; Bian et al., 2017). In particular, if a continuous function is submodular, it can also be minimized exactly in polynomial time.

The natural next question–which is addressed in this work–to ask is if submodularity still defines a boundary between easy and hard *mixed* optimization problems such as (1), where the function f in (1) is continuous, but the function g has a discrete co-domain. Our work explores this boundary and identifies sufficient conditions, based on the submodularity of both functions, under which the exact solution of problem (1) can be efficiently computed.

Exploiting submodularity in these mixed scenarios is not a new idea, given its utility in discrete optimization problems. Notable uses include establishing approximation guarantees for greedy algorithms applied to sparsity-constrained optimization (Elenberg et al., 2018), or in producing tight convex relaxations for set-function descriptions of desired sparsity patterns (Bach et al., 2012).

As highlighted above, Bach (2019) shows that if a continuous function is submodular, it can be *discretized* into a discrete submodular function, which can then be minimized exactly in polynomial time. However, this discretization is only valid for compact subsets of continuous spaces and necessarily introduces discretization error into the produced solution.

In a line of work similar to this one, authors in El Halabi and Jegelka (2020) propose converting the mixed problem to a purely discrete one without discretizing. They then advocate using a specific submodular set function minimization algorithm for solving the discrete problem, and give approximation guarantees under the assumption that the functions are nearly submodular. Our proposed approach is similar, but our work instead focuses on finding conditions under which an *arbitrary choice* (of potentially more efficient) algorithms produce *exact* results, which leads to their choice as a special case.

The sufficient conditions we require may be violated in practice. Traditionally, violations of submodularity are handled by suitably relaxing the definition with an additive or multiplicative constant and propagating the constant through a particular algorithm (El Halabi and Jegelka, 2020; Elenberg et al., 2018). Alternatively, in this work we find a sub-class of optimization problems that we can always lift into problems that satisfy our assumptions. Moreover, we prove that the solution of the lifted problem gives a near-optimal solution to the original. Our lifting approach stands in stark contrast to existing methods, as it is algorithm-independent with a guarantee that is easy to compute rather than tied to a specific algorithm and dependent on constants that are NP-Hard to compute (El Halabi and Jegelka, 2020; Elenberg et al., 2018).

We make several technical contributions, namely:

(i) We identify new sufficient conditions, based on submodularity, under which the regularized model selection problem (1) can be solved efficiently and exactly;

(ii) We extend this theory to accommodate simple continuous and discrete constraints on the model parameter for some problem classes;

(iii) We highlight the utility of exact solutions for robust optimization scenarios;

(iv) We show that problems violating our sufficient conditions can be lifted to problems that do satisfy them, and whose solutions correspond to optimal or near-optimal solutions of the original problem;

(v) We numerically validate the correctness of our theory with examples from sparse regression and retail price optimization.

2. Submodular Functions on Lattices

In this work, we consider optimization problems defined on two sets: an uncountably infinite set, typically $\mathbb{R}^n$ or a subset thereof referred to as a *continuous set*, and a countable set, typically finite and referred to as a *discrete set*. Because we would like to efficiently solve optimization problems defined on both continuous and discrete sets, we study a structure that can allow efficient optimization in both cases: submodularity.

Submodularity is typically defined as a property of set functions, which are functions that map any subset of a finite set V to a real number, i.e., $f : 2^V \to \mathbb{R}$. More generally, however, submodularity is a property of functions on *lattices* which can be continuous or discrete sets.

Let $\mathcal{X}$ be a set equipped with a partial order of its elements, denoted by $\preceq$. For any two elements $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we define their least upper bound, or *join*, as:

$$\mathbf{x} \curlyvee \mathbf{x}' = \inf\{\mathbf{y} \in \mathcal{X} \ : \ \mathbf{x} \leq \mathbf{y}, \ \mathbf{x}' \leq \mathbf{y}\}. \tag{3}$$

Dually, we define their greatest lower bound, or *meet*, as:

$$\mathbf{x} \curlywedge \mathbf{x}' = \sup\left\{\mathbf{y} \in \mathcal{X} \ : \ \mathbf{y} \leq \mathbf{x}, \ \mathbf{y} \leq \mathbf{x}'\right\}. \tag{4}$$

If for any two elements $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, their join, $\mathbf{x} \curlyvee \mathbf{x}'$, and their meet, $\mathbf{x} \curlywedge \mathbf{x}'$, exist and are in $\mathcal{X}$, then the set $\mathcal{X}$ and its order define a *lattice*. We write the lattice and its partial order together as $(\mathcal{X}, \preceq)$, but will often write just $\mathcal{X}$ when the order is clear from context. If a subset $\mathcal{S} \subseteq \mathcal{X}$ is such that for any two of its elements $\mathbf{x}, \mathbf{x}' \in \mathcal{S}$, both their join, $\mathbf{x} \curlyvee \mathbf{x}'$, and their meet, $\mathbf{x} \curlywedge \mathbf{x}'$, are in $\mathcal{S}$, the subset $\mathcal{S}$ is called a *sublattice* of $\mathcal{X}$ (Davey and Priestley, 2002).

As an example, consider a finite set of elements V. Then its power set, 2^V (the set of all its possible subsets), forms a lattice when ordered by set inclusion, $\subseteq$. Under this order, the join of any two elements $X, X' \subseteq V$ is their set union, $X \cup X' \subseteq V$, and dually, their meet is their set intersection $X \cap X' \subseteq V$.

We can also endow continuous sets with partial orders that define lattices. Recent work has brought attention to $\mathbb{R}^n$ equipped with the partial order $\preceq$, defined as:

$$\mathbf{x} \preceq \mathbf{x}' \quad \Leftrightarrow \quad \mathbf{x}_i \leq \mathbf{x}'_i \quad \text{for all } i = 1, 2, ..., n, \tag{5}$$

where $\leq$ denotes the usual order on $\mathbb{R}$.

Under this order, the join and meet operation for any two elements $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ are element-wise maximum and minimum, respectively, meaning:

$$(\mathbf{x} \curlyvee \mathbf{x}')_i = \max\{\mathbf{x}_i, \mathbf{x}'_i\}, \text{ for all } \ i = 1, 2, ..., n, \tag{6}$$

$$(\mathbf{x} \curlywedge \mathbf{x}')_i = \min\{\mathbf{x}_i, \mathbf{x}'_i\}, \text{ for all } \ i = 1, 2, ..., n. \tag{7}$$

Given a lattice $\mathcal{X}$, consider a function $f : \mathcal{X} \to \mathbb{R}$. The function f is *submodular* on the lattice $\mathcal{X}$ when the following inequality holds for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

$$f(\mathbf{x}) + f(\mathbf{x}') \geq f(\mathbf{x} \curlyvee \mathbf{x}') + f(\mathbf{x} \curlywedge \mathbf{x}'). \tag{8}$$

The function f is *monotone* when it satisfies:

$$\mathbf{x} \preceq \mathbf{x}' \quad \Longrightarrow \quad f(\mathbf{x}) \leq f(\mathbf{x}'). \tag{9}$$

When working with the lattice $(2^V, \subseteq)$, the submodular inequality (8) becomes:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \quad \text{for all } A, B \subseteq V. \tag{10}$$

Similarly, the monotonicity implication (9) becomes:

$$A \subseteq B \quad \Longrightarrow \quad f(A) \leq f(B). \tag{11}$$

Minimizing or maximizing an arbitrary set function is NP-Hard in general. If the set function is submodular, however, it can be exactly minimized and approximately maximized (up to a constant-factor approximation ratio) in polynomial time (Schrijver, 2003; Nemhauser et al., 1978). The computational tractability of submodular optimization for set functions has a variety of applications in countless fields such as sparse regression, summarization, and sensor placement (Elenberg et al., 2018; Lin and Bilmes, 2011; Krause et al., 2006).

When working with the lattice $(\mathbb{R}^n, \preceq)$, a function $f : \mathbb{R}^n \to \mathbb{R}$ is submodular when:

$$f(\mathbf{x}) + f(\mathbf{x}') \geq f(\max\{\mathbf{x}, \mathbf{x}'\}) + f(\min\{\mathbf{x}, \mathbf{x}'\}) \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^n, \tag{12}$$

where the maximum and minimum operations are performed element-wise, as expressed in (6) and (7). When f is twice differentiable, submodularity on $\mathbb{R}^n$ is equivalent (see Topkis 1998; Bach 2019) to the condition:

$$\frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \leq 0 \quad \text{for all } i \neq j. \tag{13}$$

Perhaps surprisingly, the guarantees associated with submodular set function optimization extend to functions that are submodular on $\mathbb{R}^n$. In particular, submodular functions on $\mathbb{R}^n$ can be minimized over a bounded sublattice in polynomial time (see Bach 2019), and can be approximately maximized with constant-factor approximation ratios (Bian et al., 2016, 2017).

3. Problem Formulation

In this section, we bridge continuous and discrete submodular function minimization in one unified problem statement. We do this by drawing inspiration from the field of structured sparsity, where the choice of zero entries in real-valued decision variables is viewed as a coupled discrete and continuous problem (Bach, 2013, 2011).

To highlight the connection with structured sparsity problems, for $n \in \mathbb{Z}_{>0}$, we denote by $[n]$ the set $\{1, 2, ..., n\}$, and by $2^{[n]}$ the set of all possible subsets of $[n]$. Define the map $\text{supp} : \mathbb{R}^n \to 2^{[n]}$ as:

$$\text{supp}\,(\mathbf{x}) = \{i \in [n] \mid \mathbf{x}_i \neq 0\}. \tag{14}$$

In words, supp returns the set of indices where the vector $\mathbf{x}$ is nonzero. Consider arbitrary functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : 2^{[n]} \to \mathbb{R}$. Problems of the form:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(\text{supp}\,(\mathbf{x})), \tag{15}$$

often arise in structured sparse optimization, where the preferences in discrete selections (the zero entries of $\mathbf{x}$) are expressed through the function g. As a special case, if we let $f(\mathbf{x}) = \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ with $\mathbf{D} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ and define $g(A) = |A|$ as the cardinality of the set A, (15) becomes:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2 + \|\mathbf{x}\|_0, \tag{CS}$$

where $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm. The problem (CS) is a form of the well-studied compressed sensing problem, which is NP-Hard in general (Rauhut, 2010).

Generalizing the idea of making continuous decisions through the choice of $\mathbf{x}$ in (15), and discrete decisions through the choice of the zero entries of $\mathbf{x}$, we consider two lattices, $(\mathcal{X}, \preceq)$ and $(\mathcal{Y}, \sqsubseteq)$, related by a map $\eta : \mathcal{X} \to \mathcal{Y}$. We let $f : \mathcal{X} \to \mathbb{R}$ be a function describing the cost of assignments of variables in $\mathcal{X}$, and similarly let $g : \mathcal{Y} \to \mathbb{R}$ describe the associated cost of choices in $\mathcal{Y}$. Then, we seek the optimal point $\mathbf{x}^* \in \mathcal{X}$ in the problem:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}} \quad f(\mathbf{x}) + g(\eta(\mathbf{x})). \tag{P}$$

Although we will eventually let $\mathcal{X}$ describe continuous choices and $\mathcal{Y}$ describe associated discrete ones, our theoretical results do not rely on the cardinality of the lattices $\mathcal{X}$ and $\mathcal{Y}$.

Intuitively, problem (P) asks for the element $\mathbf{x} \in \mathcal{X}$ which incurs minimum cost in $\mathcal{X}$, as measured by $f(\mathbf{x})$, and in $\mathcal{Y}$, as measured by $g(\eta(\mathbf{x}))$. Given that the special case of (CS) is already hard in general, with no additional structure on f, g and η, this problem is hopelessly difficult. To provide the necessary structure, we make the following assumptions.

Assumptions 1 *Consider the lattices $(\mathcal{X}, \preceq)$ and $(\mathcal{Y}, \sqsubseteq)$ and the maps $\eta : \mathcal{X} \to \mathcal{Y}$, $f : \mathcal{X} \to \mathbb{R}$, and $g : \mathcal{Y} \to \mathbb{R}$. We make the following assumptions:*

1. *The functions f and g are submodular on the lattices $\mathcal{X}$ and $\mathcal{Y}$, respectively,*
2. *The function g is monotone on $\mathcal{Y}$,*
3. *For all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:*

$$\eta(\mathbf{x} \curlyvee \mathbf{x}') \sqsubseteq \eta(\mathbf{x}) \sqcup \eta(\mathbf{x}'), \quad \eta(\mathbf{x} \curlywedge \mathbf{x}') \sqsubseteq \eta(\mathbf{x}) \sqcap \eta(\mathbf{x}').$$

Remark 1 *If the map $\eta : \mathcal{X} \to \mathcal{Y}$ satisfies Assumption 3, it is an order-preserving join-homomorphism, meaning it maintains the order and joins of elements in $\mathcal{X}$. (Prop. 2.19 in Davey and Priestley 2002) Explicitly, Assumption 3 is equivalent to the condition that for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:*

$$\mathbf{x} \preceq \mathbf{x}' \Rightarrow \eta(\mathbf{x}) \sqsubseteq \eta(\mathbf{x}'),$$
$$\eta(\mathbf{x} \curlyvee \mathbf{x}') = \eta(\mathbf{x}) \sqcup \eta(\mathbf{x}').$$

Despite this equivalence, we leave Assumption 3 as written above for clarity in future proofs.

We highlighted the lattices $(\mathbb{R}^n, \preceq)$ and $(2^{[n]}, \subseteq)$, but for the map $\text{supp} : \mathbb{R}^n \to 2^{[n]}$ to satisfy Assumption 3, we must restrict the domain of f to only only the first orthant, $(\mathbb{R}^n_{\geq 0}, \preceq)$. As mentioned by Bian et al. (2017), this issue can often be resolved by considering an appropriate *orthant conic lattice*, which views $\mathbb{R}^n$ as a product of n copies of $\mathbb{R}$ and selects a different order for each copy. Alternatively, any least-squares problem such as (CS) can be lifted to a non-negative least-squares problem, allowing us to satisfy Assumption 3 with the map supp, but potentially no longer satisfying Assumption 1 (see Appendix A).

Assumption 1, which requires f and g to be submodular can be restrictive in practice. To mitigate this, in Section 7 we show how some specific problem instances that do not satisfy Assumption 1–in particular when f is quadratic–can be lifted to a new optimization problem that satisfies all the required assumptions. We then derive conditions under which solving the new, lifted problem still provides a solution to the original problem that violated Assumption 1. In contrast, the more typical way of handling non-submodular f involves relaxing the definition of submodularity (8) to include an additive or multiplicative constant and propagating it through a chosen algorithm to give near-optimality guarantees. El Halabi and Jegelka (2020); Elenberg et al. (2018) Our suggested lifting, however, sidesteps the need for a particular algorithm while still providing optimality or near-optimality guarantees.

4. Solving an Equivalent Problem

In this section, we outline our approach for solving the problem (P) by defining a related optimization problem on a single lattice. We then prove that this related problem is a submodular function minimization problem, and that by solving it we recover a solution to (P). Finally, we highlight some conditions under which solving this related problem is a polynomial time operation.

4.1 The Equivalent Submodular Minimization Problem

As expressed above, the problem (P) asks for the a choice of $\mathbf{x} \in \mathcal{X}$ and associated $\eta(\mathbf{x}) \in \mathcal{Y}$. Our key observation is that we could instead ask for a choice of $\mathbf{y} \in \mathcal{Y}$ and best associated $\mathbf{x} \in \mathcal{X}$, leading to the problem:

$$\underset{\mathbf{y} \in \mathcal{Y}}{\text{minimize}} \quad g(\mathbf{y}) + \min_{\substack{\mathbf{x} \in \mathcal{X} \\ \eta(\mathbf{x}) = \mathbf{y}}} f(\mathbf{x}).$$

In the special case of (CS) explored earlier, this equivalent problem becomes:

$$\underset{S \in 2^{[n]}}{\text{minimize}} \quad |S| + \min_{\substack{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \\ \text{supp}(\mathbf{x}) = S}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

While this new problem is clearly the same as (CS), the innermost minimization is over the set of $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ such that $\text{supp}\,(\mathbf{x}) = S$, or equivalently, $\mathbf{x}_i \neq 0$ for all $i \in S$, and $\mathbf{x}_i = 0$ for all $i \notin S$. This feasible set is not a closed subset of $\mathbb{R}^n_{\geq 0}$, and thus the corresponding minimizer of this innermost problem may not exist (Borwein and Lewis, 2006).

With this issue in mind, we instead consider a slight relaxation of the above problem:

$$\underset{\mathbf{y}\in\mathcal{Y}}{\text{minimize}} \;\; g(\mathbf{y}) + H(\mathbf{y}), \tag{P-R}$$

where we have defined the function $H : \mathcal{Y} \to \mathbb{R}$ as:

$$H(\mathbf{y}) = \min_{\substack{\mathbf{x}\in\mathcal{X} \\ \eta(\mathbf{x})\sqsubseteq\mathbf{y}}} f(\mathbf{x}). \tag{16}$$

In the special case of (CS), this relaxation produces the problem:

$$\underset{S\in 2^{[n]}}{\text{minimize}} \;\; |S| + \min_{\substack{\mathbf{x}\in\mathbb{R}^n_{\geq 0} \\ \text{supp}(\mathbf{x})\subseteq S}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2, \tag{CS-R}$$

where the innermost minimization is instead over the set of $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ such that $\mathbf{x}_i = 0$ for all $i \notin S$, which is a closed subset of $\mathbb{R}^n_{\geq 0}$.

We now prove that under Assumptions 1-3, the relaxed problem (P-R) is a submodular minimization problem, and that by solving it we can recover the corresponding minimizer for (P). As established above, minimizing functions on finitely presentable distributive lattices is efficient when the functions are submodular, so we show that the relaxed problem (P-R) is a submodular function minimization problem on $\mathcal{Y}$.

Theorem 2 *Under Assumptions 1-3, the function $g + H : \mathcal{Y} \to \mathbb{R}$ is submodular on $\mathcal{Y}$, and therefore the relaxed problem* (P-R) *is a submodular function minimization problem over $\mathcal{Y}$. Moreover, let $\mathbf{y}^* \in \mathcal{Y}$ be the minimizer for the problem* (P-R)*, and let $\mathbf{x}^* \in \mathcal{X}$ be such that:*

$$\mathbf{x}^* \in \underset{\substack{\mathbf{x}\in\mathcal{X} \\ \eta(\mathbf{x})\sqsubseteq\mathbf{y}^*}}{\text{argmin}} \; f(\mathbf{x}).$$

Then $\mathbf{x}^*$ is a minimizer for the problem (P)*.*

To prove this result, we require a few technical lemmas.

Lemma 3 *Let $(\mathcal{X}, \preceq)$ and $(\mathcal{Y}, \sqsubseteq)$ be lattices with the map $\eta : \mathcal{X} \to \mathcal{Y}$ satisfying Assumption 3. Then the set:*

$$\mathcal{D} = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \mid \eta(\mathbf{x}) \sqsubseteq \mathbf{y}\}, \tag{17}$$

is a sublattice of the product lattice, $\mathcal{X} \times \mathcal{Y}$.

Proof On the product lattice, the join of any two elements $(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \mathcal{D}$ is denoted by $\vee_{\mathcal{D}}$, and defined as:

$$(\mathbf{x}, \mathbf{y}) \vee_{\mathcal{D}} (\mathbf{x}', \mathbf{y}') = (\mathbf{x} \curlyvee \mathbf{x}', \mathbf{y} \sqcup \mathbf{y}').$$

Then, we note that for this same $(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \mathcal{D}$:

$$\eta(\mathbf{x} \curlyvee \mathbf{x}') \sqsubseteq \eta(\mathbf{x}) \sqcup \eta(\mathbf{x}') \sqsubseteq \mathbf{y} \sqcup \mathbf{y}',$$

where we first used Assumption 3, then the fact that $(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}') \in \mathcal{D}$. Therefore, the pair $(\mathbf{x} \curlyvee \mathbf{x}', \mathbf{y} \sqcup \mathbf{y}')$ is also in D.

Because $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ were arbitrary, this holds for all of $\mathcal{D}$. A dual analysis follows for the meet operation. ■

The sublattice $\mathcal{D}$ is useful as the only pairs of $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ considered in the problem (P-R) are those that are in $\mathcal{D}$. The following theorem then uses this sublattice to prove that H is submodular. The result is a simple application of an established theorem in literature, but we include its proof here for completeness.

Theorem 4 *(Application of Theorem 2.7.6 in Topkis 1998) Let $f : \mathcal{X} \to \mathbb{R}$, $g : \mathcal{Y} \to \mathbb{R}$, and $\eta : \mathcal{X} \to \mathcal{Y}$ be maps satisfying Assumptions 1 and 3. Then the function $g + H : \mathcal{Y} \to \mathbb{R}$, with H defined as in* (16)*, is submodular on $\mathcal{Y}$.*

Proof To prove this statement, we take two points $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$ and compare the values of the function $g + H$, verifying the submodular inequality (8). We note that for any $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, there are corresponding $\mathbf{z}, \mathbf{z}' \in \mathcal{X}$ such that:

$$\begin{aligned} \mathbf{z} \in \operatorname*{argmin}_{\substack{\mathbf{x} \in \mathcal{X} \\ \eta(\mathbf{x}) \sqsubseteq \mathbf{y}}} f(\mathbf{x}) \quad &\Rightarrow \quad H(\mathbf{y}) = f(\mathbf{z}), \\ \mathbf{z}' \in \operatorname*{argmin}_{\substack{\mathbf{x} \in \mathcal{X} \\ \eta(\mathbf{x}) \sqsubseteq \mathbf{y}'}} f(\mathbf{x}) \quad &\Rightarrow \quad H(\mathbf{y}') = f(\mathbf{z}'). \end{aligned} \tag{18}$$

By definition, $(\mathbf{z}, \mathbf{y})$ and $(\mathbf{z}', \mathbf{y}')$ are both in the subset $\mathcal{D}$ as defined in (17). Then, it follows:

$$\begin{aligned} g(\mathbf{y}) + H(\mathbf{y}) + g(\mathbf{y}') + H(\mathbf{y}') &= g(\mathbf{y}) + f(\mathbf{z}) + g(\mathbf{y}') + f(\mathbf{z}') \\ &\geq g(\mathbf{y} \sqcup \mathbf{y}') + g(\mathbf{y} \sqcap \mathbf{y}') + f(\mathbf{z} \curlyvee \mathbf{z}') + f(\mathbf{z} \curlywedge \mathbf{z}'), \end{aligned}$$

where we first used (18) and then the submodularity of f and g.

By Lemma 3, $\mathcal{D}$ is a sublattice of $\mathcal{X} \times \mathcal{Y}$, and so the pairs $(\mathbf{z} \curlyvee \mathbf{z}', \mathbf{y} \sqcup \mathbf{y}')$ and $(\mathbf{z} \curlywedge \mathbf{z}', \mathbf{y} \sqcap \mathbf{y}')$ are also in $\mathcal{D}$, meaning:

$$\begin{aligned} \eta(\mathbf{z} \curlyvee \mathbf{z}') &\sqsubseteq \mathbf{y} \sqcup \mathbf{y}', \\ \eta(\mathbf{z} \curlywedge \mathbf{z}') &\sqsubseteq \mathbf{y} \sqcap \mathbf{y}'. \end{aligned}$$

Therefore $\mathbf{z} \curlyvee \mathbf{z}'$ and $\mathbf{x} \curlywedge \mathbf{x}'$ are feasible points in the minimization defining $H(\mathbf{y} \sqcup \mathbf{y}')$ and $H(\mathbf{y} \sqcap \mathbf{y}')$, respectively, in (16). We then have, as desired:

$$\begin{aligned} g(\mathbf{y}) + H(\mathbf{y}) + g(\mathbf{y}') + H(\mathbf{y}') &\geq g(\mathbf{y} \sqcup \mathbf{y}') + g(\mathbf{y} \sqcap \mathbf{y}') + f(\mathbf{z} \curlyvee \mathbf{z}') + f(\mathbf{z} \curlywedge \mathbf{z}') \\ &\geq g(\mathbf{y} \sqcup \mathbf{y}') + g(\mathbf{y} \sqcap \mathbf{y}') + \min_{\substack{\mathbf{x} \in \mathcal{X} \\ \eta(\mathbf{x}) \sqsubseteq \mathbf{y} \sqcup \mathbf{y}'}} f(\mathbf{x}) + \min_{\substack{\mathbf{x} \in \mathcal{X} \\ \eta(\mathbf{x}) \sqsubseteq \mathbf{y} \sqcap \mathbf{y}'}} f(\mathbf{x}) \\ &= g(\mathbf{y} \sqcup \mathbf{y}') + H(\mathbf{y} \sqcup \mathbf{y}') + g(\mathbf{y} \sqcap \mathbf{y}') + H(\mathbf{y} \sqcap \mathbf{y}'). \end{aligned}$$

■

Because $g + H$ is submodular on $\mathcal{Y}$, solving (P-R), is an instance of submodular function minimization. What remains is to show that solving this relaxed problem allows us to also solve to the original problem, (P).

Lemma 5 *Let $\mathbf{y}^* \in \mathcal{Y}$ be a minimizer for the relaxed problem* (P-R)*, and let $\mathbf{x}^* \in \mathcal{X}$ be such that:*

$$\mathbf{x}^* \in \underset{\substack{\mathbf{x}\in\mathcal{X}\\ \eta(\mathbf{x})\sqsubseteq\mathbf{y}^*}}{\operatorname{argmin}} f(\mathbf{x}).$$

If g satisfies Assumption 2, then $\mathbf{x}^*$ is a minimizer for the problem (P)*.*

Proof To prove this lemma, we consider an optimal $\mathbf{z}^* \in \mathcal{X}$ for problem (P) and verify that the proposed minimizer, $\mathbf{x}^* \in \mathcal{X}$, has the same cost.

We first note that by the optimality of $\mathbf{z}^*$ in problem (P):

$$f(\mathbf{z}^*) + g(\eta(\mathbf{z}^*)) \leq f(\mathbf{x}^*) + g(\eta(\mathbf{x}^*)). \tag{19}$$

Additionally, we have:

$$\begin{aligned} f(\mathbf{z}^*) + g(\eta(\mathbf{z}^*)) &\geq \min_{\substack{\mathbf{x}\in\mathcal{X}\\ \eta(\mathbf{x})\sqsubseteq\eta(\mathbf{z}^*)}} f(\mathbf{x}) + g(\eta(\mathbf{z}^*)) && \text{(minimizing, as } \mathbf{z}^* \text{ is feasible)} \\ &= H(\eta(\mathbf{z}^*)) + g(\eta(\mathbf{z}^*)) && \text{(definition of } H) \\ &\geq H(\mathbf{y}^*) + g(\mathbf{y}^*) && \text{(optimality of } \mathbf{y}^* \text{ in P-R)} \\ &= f(\mathbf{x}^*) + g(\mathbf{y}^*) && \text{(definition of } \mathbf{x}^*). \end{aligned}$$

This sequence of inequalities implies:

$$f(\mathbf{z}^*) + g(\eta(\mathbf{z}^*)) \geq f(\mathbf{x}^*) + g(\mathbf{y}^*). \tag{20}$$

Note that because g is monotone, $g(\mathbf{y}^*) \geq g(\eta(\mathbf{x}^*))$. Using this fact, we can lower bound the right-hand side of (20):

$$f(\mathbf{z}^*) + g(\eta(\mathbf{z}^*)) \geq f(\mathbf{x}^*) + g(\mathbf{y}^*) \geq f(\mathbf{x}^*) + g(\eta(\mathbf{x}^*)).$$

By the optimality of $\mathbf{z}^*$, we see that $\mathbf{x}^*$ must also be optimal for the problem (P). ■

This series of results gives rise to Theorem 2, which provides sufficient conditions under which we can transform problem (P), an optimization problem on two lattices, into problem (P-R), a submodular function minimization problem on a single lattice.

Proof *(Theorem 2)*
Under Assumptions 1 and 3, Theorem 4 states that the function $g + H : \mathcal{Y} \to \mathbb{R}$ is submodular on the lattice $\mathcal{Y}$. Therefore, solving (P-R) is a submodular function minimization problem over $\mathcal{Y}$, and the first part of the theorem is proved.

Under Assumption 2, by Lemma 5, given the minimizer $\mathbf{y}^*$ of (P-R), the point $\mathbf{x}^* \in \mathcal{X}$ defined by:

$$\mathbf{x}^* \in \underset{\substack{\mathbf{x}\in\mathcal{X}\\ \eta(\mathbf{x})\sqsubseteq\mathbf{y}^*}}{\operatorname{argmin}} f(\mathbf{x}),$$

is a minimizer in the original problem (P). ■

4.2 Solving (P-R) in Polynomial Time

Despite the submodular structure of the functions, we can only truly solve (P-R) in polynomial time if $\mathcal{Y}$ is a finitely presentable distributive lattice and we have an oracle for evaluating the functions g and H, which we formally state next.

Corollary 6 *Let $f : \mathcal{X} \to \mathbb{R}$ be a submodular function on $(\mathcal{X}, \preceq)$, let $(\mathcal{Y}, \sqsubseteq)$ be a finitely presentable distributive or diamond modular lattice with $g : \mathcal{Y} \to \mathbb{R}$ a monotone submodular function, and let $\eta : \mathcal{X} \to \mathcal{Y}$ satisfy Assumption 3. If we have access to an evaluation oracle for $g + H$, then problem* (P) *can be solved in a polynomial number of operations and a polynomial number of calls to the oracle.*

Proof Assumptions 1, 2, and 3 are satisfied, by $\mathcal{X}$, $\mathcal{Y}$, and the functions η, f, and g. By Theorem 2, therefore, we can solve the problem (P) by instead minimizing $g + H$ over $\mathcal{Y}$, i.e., solving problem (P-R). Problem (P-R) is a submodular function minimization problem over a a finitely presentable distributive or diamond modular lattice, which established algorithms can solve in a polynomial number of operations and oracle calls to $g+H$ (Fujishige et al., 2022; Schrijver, 2003). ■

With Corollary 6 in hand, we need to construct the required oracle for $H : \mathcal{Y} \to \mathbb{R}$ that only requires a polynomial number of operations. Once we have an oracle for H (assuming another oracle or polynomial algorithm for evaluating g), solving (P) clearly only requires a polynomial number of operations.

We are particularly interested in joint continuous and discrete optimization, such as when the relevant lattices are $(\mathcal{X}, \preceq) = (\mathbb{R}^n_{\geq 0}, \sqsubseteq)$ and $(\mathcal{Y}, \preceq) = (2^{[n]}, \subseteq)$ connected by the map $\operatorname{supp} : \mathbb{R}^n_{\geq 0} \to 2^{[n]}$ as expressed in (14). In this case, evaluating H requires solving the optimization problem:

$$\begin{aligned} &\underset{\substack{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \\ \operatorname{supp}(\mathbf{x}) \subseteq A}}{\text{minimize}} \; f(\mathbf{x}), \end{aligned} \tag{21}$$

for any $A \in 2^{[n]}$.

As discussed above, when $\mathcal{X}$ is the product of bounded intervals, we can rely on the continuous submodular minimization algorithms developed by Bach (2019). These algorithms, however, introduce discretization error, limiting the accuracy of the evaluations of H. Moreover, the simple example of (21) is a continuous submodular minimization problem, but the set $\mathbb{R}_{\geq 0}$ is not a bounded sublattice and thus the algorithms of Bach (2019) do not directly apply. Continuous submodularity alone appears limited in this way, so we pursue other problem structures leading to algorithms for efficient and arbitrarily accurate solutions of (21).

Note that for any $A \in 2^{[n]}$, the feasible set for the sub-problem (21) is a convex subset of $\mathbb{R}^n_{\geq 0}$. If the function $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ is convex, under appropriate regularity conditions, we can use any generic convex optimization routine to solve the associated sparsity-constrained problem (21). For example, in the compressed sensing scenario shown in (CS-R), evaluating H amounts to solving a simple reduced least-squares problem. More generally, we need $f : \mathcal{X} \to \mathbb{R}$ to be convex and submodular, and the set of $\mathbf{x} \in \mathcal{X}$ such that $\eta(\mathbf{x}) \sqsubseteq \mathbf{y}$ to be a

compact, convex subset for every $\mathbf{y} \in \mathcal{Y}$, alongside sufficient regularity conditions, such as constraint qualifications or the existence of separation oracles (Borwein and Lewis, 2006; Schrijver, 2003).

We have already assumed that f is submodular (in this case, on $\mathbb{R}^n_{\geq 0}$), but submodular functions are neither a subset nor a superset of convex functions, so we may also require that f is convex. For example, any separable convex function f satisfies this assumption, as do convex quadratic functions with non-positive off-diagonal entries, or functions on $\mathbb{R}^n$ that can be identified as the Lovász extension of submodular *set* functions.

Our theory is completely agnostic to the choice of algorithms for both evaluating H and solving the discrete optimization problem (P-R). In particular, if we assume f is convex, evaluate it through convex optimization, and use projected subgradient descent on the Lovàsz extension of $g + H$ as the algorithm for solving the set function minimization, we recover exactly the approach proposed by El Halabi and Jegelka (2020).

Convexity of f is not the only additional assumption on f that leads to tractable evaluations of H without resorting to continuous submodular minimization algorithms. As an alternative, we could consider a nonconvex quadratic form for $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{p}^T \mathbf{x}, \tag{22}$$

with $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{p} \in \mathbb{R}^n$. The assumption that this quadratic function is submodular on $\mathbb{R}^n_{\geq 0}$ is equivalent to the condition:

$$\frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \mathbf{Q}_{ij} \leq 0, \quad \text{for all } i \neq j.$$

Moreover, for a given $A \in 2^{[n]}$, our sub-problem instance (21) is a constrained, nonconvex quadratic program:

$$\begin{array}{ll} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} & \mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{p}^T \mathbf{x} \\ \text{subject to} & \mathbf{x} \geq 0 \\ & \mathbf{x}_i = 0, \ i \notin A. \end{array} \tag{23}$$

Researchers Kim and Kojima (2003) have established that nonconvex quadratic programs satisfying submodularity admit tight semidefinite program relaxations. In particular, we have the following theorem:

Theorem 7 *(Theorem 3.1 in Kim and Kojima 2003) Let $\mathbf{Q} \in \mathbb{R}^{n \times n}$ have nonpositive off-diagonal entries. Let $\mathrm{tr} : \mathbb{R}^{n \times n} \to \mathbb{R}$ denote the trace of a matrix, $\mathrm{diag} : \mathbb{R}^{n \times n} \to \mathbb{R}^n$ denote the diagonal entries of the matrix, and let $\succeq$ indicate the positive semidefiniteness of a symmetric matrix. Further, for any $A \in 2^{[n]}$, let $\mathbf{Z}_{A^c}$ denote the rows and columns of $\mathbf{Z}$ with indices not in the set A. Consider the semi-definite program:*

$$\begin{array}{ll} \underset{\substack{\mathbf{z} \in \mathbb{R}^n \\ \mathbf{Z} \in \mathbb{S}^n}}{\textit{minimize}} & \mathrm{tr}\,(\mathbf{Q}\mathbf{Z}) + 2\mathbf{p}^T \mathbf{z} \\ \textit{subject to} & \mathrm{tr}\,(\mathbf{Z}_{A^c}) \leq 0 \\ & \mathrm{diag}\,(\mathbf{Z}) \geq 0 \\ & \begin{bmatrix} 1 & \mathbf{z}^T \\ \mathbf{z} & \mathbf{Z} \end{bmatrix} \succeq 0, \end{array}$$

Given the solution $(\mathbf{Z}^*, \mathbf{z}^*)$ *to this SDP, the vector* $\mathbf{x}_i^* = \sqrt{\mathbf{Z}_{ii}^*}$, $i = 1, ..., n$ *is a minimizer for the non-convex quadratic program* (23).

Because semi-definite programs satisfying appropriate constraint qualifications can be solved in polynomial time, we could use this relaxation to evaluate H for any subset $A \in 2^{[n]}$. This approach produces the required oracle for Corollary 6, but only requires that quadratic functions f of the form (22) satisfy submodularity.

5. Constrained Optimization

In this and the following sections, we extend our framework both theoretically and algorithmically for the specific case of the lattices $(\mathbb{R}^n_{\geq 0}, \preceq)$ and $(2^{[n]}, \subseteq)$, connected by the support map $\text{supp} : \mathbb{R}^n_{\geq 0} \to 2^{[n]}$.

In many problems, we may be interested in optimization over a feasible strict subset $C \subset \mathbb{R}^n_{\geq 0}$. Unfortunately, submodular function minimization and maximization subject to constraints is NP-Hard in general (Fujishige and Isotani, 2011). This difficulty arises because arbitrary subsets of a lattice rarely define sublattices.

One simple class of problems whose feasible sets are not sublattices are problems with *budget constraints*:

$$\begin{array}{ll} \underset{\mathbf{x} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} & f(\mathbf{x}) + g(\text{supp}\,(\mathbf{x})) \\ \text{subject to} & \sum_{i=1}^n W_i(\mathbf{x}_i) \leq B, \end{array} \tag{24}$$

with $W_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ strictly increasing functions for $i = 1, 2, ..., n$ and $B \in \mathbb{R}_{>0}$ a "budget".

When confronted with constrained optimization problems such as (24), one common approach is to add a Lagrange multiplier $\mu \in \mathbb{R}_{\geq 0}$ and instead solve the unconstrained problem:

$$\underset{\mathbf{x} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad f(\mathbf{x}) + g(\text{supp}\,(\mathbf{x})) + \mu \sum_{i=1}^n W_i(\mathbf{x}_i). \tag{25}$$

For the correct choice of $\mu \in \mathbb{R}_{\geq 0}$, solving the regularized problem (25) can be equivalent to solving the constrained problem (24) (Nagano et al., 2011; Staib and Jegelka, 2019). Because (24) is non-convex, identifying when this approach is valid requires some careful detail. When possible, however, determining the μ that renders the two problems equivalent is typically a difficult task.

Our work in this section relies on the following result that relates parameterized families of submodular set function minimization problems to a single convex optimization problem.

Theorem 8 *(Proposition 8.4 in Bach 2013) Let* $h : 2^{[n]} \to \mathbb{R}$ *be a submodular set function, and* $h_L : \mathbb{R}^n \to \mathbb{R}$ *its Lovàsz extension (which is therefore convex). If, for some* $\epsilon > 0$, $\psi_i : \mathbb{R}_{\geq \epsilon} \to \mathbb{R}$ *is a strictly increasing function on its domain for all* $i = 1, 2, ..., n$, *then the minimizer* $\mathbf{u}^* \in \mathbb{R}^n_{\geq 0}$ *of the convex optimization problem:*

$$\underset{\mathbf{u} \in \mathbb{R}^n_{\geq 0}}{\textit{minimize}} \quad h_L(\mathbf{u}) + \sum_{i=1}^n \int_{\epsilon}^{\epsilon + \mathbf{u}_i} \psi_i(\mu) d\mu, \tag{26}$$

is such that the set $A^\mu = \{i \in [n] \ : \ \mathbf{u}_i^* > \mu\}$ *is the minimizer with smallest cardinality for the submodular set function minimization problem:*

$$\underset{A \in 2^{[n]}}{\text{minimize}} \quad h(A) + \sum_{i \in A} \psi_i(\mu), \tag{27}$$

for any $\mu \in \mathbb{R}_{\geq \epsilon}$.

In the following subsections we identify classes of problems that allow the regularized problem (25) to be expressed in the form given by (27). Theorem 8 then provides a single convex optimization problem we can solve to recover the solution to (25) for all possible values of the regularization strength μ. In prior work, this same theory was applied to purely discrete submodular minimization problems (Fujishige and Isotani, 2011), and purely continuous submodular minimization problems (Staib and Jegelka, 2019), but our work lies between these two extremes.

5.1 Support Knapsack Constraints

We first consider a knapsack constraint, meaning the function W has the form:

$$W(\mathbf{x}) = \sum_{j \in \text{supp}(\mathbf{x})} \mathbf{w}_j,$$

for some $\mathbf{w} \in \mathbb{R}^n_{>0}$. The regularized problem (25) in this case is:

$$\underset{\mathbf{x} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad f(\mathbf{x}) + g(\text{supp}\,(\mathbf{x})) + \mu \sum_{j \in \text{supp}(\mathbf{x})} \mathbf{w}_j.$$

Because W is a set function in this case, the relaxed problem (P-R) becomes:

$$\underset{A \in 2^{[n]}}{\text{minimize}} \quad g(A) + H(A) + \sum_{j \in A} \psi_j(\mu), \tag{28}$$

where we have defined $\psi_j(\mu) = \mu \mathbf{w}_j$ for each $j = 1, 2, ..., n$. Because $\mathbf{w}_j > 0$ for all j, these functions are strictly increasing, and we have a problem in the form (27). By Theorem 8, we can solve the convex optimization problem:

$$\underset{\mathbf{u} \in \mathbb{R}_{\geq \epsilon}}{\text{minimize}} \quad g_L(\mathbf{u}) + H_L(\mathbf{u}) + \frac{1}{2} \sum_{j=1}^{n} \mathbf{w}_j \mathbf{u}_j^2,$$

then appropriately threshold the solution to recover the solution to (28) for all possible values of $\mu \in \mathbb{R}_{\geq \epsilon}$. Because ψ_j is finite and strictly increasing on all of $\mathbb{R}$, we can simply select $\epsilon = 0$.

Given the solutions to the regularized problem A^μ specified by Theorem 8, we select the set A^μ with smallest $\mu \in \mathbb{R}$ such that the constraint $W(\mathbf{x}) \leq B$ is satisfied. Note however, that we only recover the solution for *any* given $B \in \mathbb{R}_{\geq 0}$ if the elements of $\mathbf{u}^*$ are unique (Bach, 2013). Otherwise, we only recover the solutions for a few particular values of B. If these elements are unique, however, we can use the result of Theorem 2 to compute the minimizer in the original optimization problem over $\mathbb{R}^n_{\geq 0}$. Moreover, by the same argument as in (Nagano et al., 2011), this solution corresponds to the solution of the original constrained problem.

5.2 Continuous Budget Constraints

As shown above, the Lovàsz extension lets us handle problems with discrete budget constraints, so a natural next step is to consider continuous budget constraints, meaning continuous functions $W : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$, such that:

$$W(\mathbf{x}) = \sum_{i=1}^{n} W_i(\mathbf{x}_i),$$

with each $W_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ a strictly increasing function. With this particular W, the regularized optimization problem (25) with Lagrange multiplier $\mu \in \mathbb{R}_{\geq 0}$ becomes:

$$\underset{\mathbf{x} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad f(\mathbf{x}) + g(\text{supp}\,(\mathbf{x})) + \mu \sum_{i=1}^{n} W_i(\mathbf{x}_i).$$

To recover the problem form (27) specified by Theorem 8, we further assume that $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ is separable, i.e., $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i)$. In this case, the relaxed optimization problem (P-R) is:

$$\underset{A \in 2^{[n]}}{\text{minimize}} \quad g(A) + \sum_{i \in A} H_i(\mu), \tag{29}$$

where we defined $H_i : \mathbb{R}_{>0} \to \mathbb{R}$ as the function:

$$H_i(\mu) = \min_{\mathbf{z} \geq 0} \; f_i(\mathbf{z}) + \mu W_i(\mathbf{z}), \quad i = 1, 2, ..., n, \tag{30}$$

and assumed (without loss of generality) that $W_i(0) = f_i(0) = 0$.

To apply Theorem 8, we need $H_i : \mathbb{R}_{>0} \to \mathbb{R}$ to be strictly increasing on its domain. We verify this property in the following proposition, whose proof we detail in Appendix B.

Proposition 9 *The function $H_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\leq 0}$ defined in (30) is monotone in μ for all $i = 1, 2..., n$. It is strictly increasing for all $\mu \in [0, c]$, where $c \in \mathbb{R}_{\geq 0}$ is the smallest constant such that $H_i(c) = 0$. In addition, H_j is constant and zero on the interval $[c, \infty[$.*

Because the only point at which H_i is not strictly increasing occurs when its value is exactly zero (implying that allowing the element $\mathbf{x}_i$ to be nonzero provides no decrease in continuous cost), the desired result from Theorem 8 still holds with only a minor modification, the details of which we also defer to Appendix B.

It then follows from Theorem 8 that by solving the single convex optimization problem:

$$\underset{\mathbf{u} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad g_L(\mathbf{u}) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon + \mathbf{u}_i} H_i(\mu) d\mu, \tag{31}$$

we can recover the solution to a family of regularized optimization problems (29). As before, we select the set A^μ with the largest $\mu \in \mathbb{R}_{\geq \epsilon}$ such that the budget constraint $W(\mathbf{x}) \leq B$ is satisfied. As discussed above, we only recover the solution for *all* $B \in \mathbb{R}_{\geq 0}$ if the elements of $\mathbf{u}^*$ are all unique. Within each choice of support, simple convex duality–which we can apply when f_i and W_i are convex functions–guarantees the existence of a $\mu \in \mathbb{R}_{\geq 0}$ that renders the constrained problem and the regularized problem equivalent.

6. Robust Optimization

Joint continuous and discrete optimization problems can easily arise as sub-problems in larger contexts. For example, in *robust optimization*, we seek to solve an optimization problem while remaining resilient to worst-case problem instances.

6.1 Motivating Example from Multiple Domain Learning

Recent work by Qian et al. (2019) highlighted the concept of *multiple domain learning*, where a single machine learning model is trained on sets of data from K different domains. By training against worst-case distributions of the data in these domains, they show that the resulting machine learning model often achieves lower generalization and worst-case testing errors.

In particular, let the training data for a learning model be $S = \{S_1, S_2, ..., S_K\}$ with S_i the data from domain i. We also let $f_i : W \to \mathbb{R}$ for $i = 1, 2, ..., K$ be the empirical risk of the model on the data from each domain i, given parameters in some convex subset $W \subseteq \mathbb{R}^n$. The proposed robust optimization problem is then:

$$\underset{\mathbf{w}\in W}{\text{minimize}} \ \max_{\mathbf{p}\in C} \ \sum_{i=1}^{K} \mathbf{p}_i f_i(\mathbf{w}),$$

with $C = \{\mathbf{p} \in \mathbb{R}^K_{\geq 0} \mid \sum_{i=1}^K \mathbf{p}_i \leq 1\}$, the simplex. If we additionally reward the use of data from domain i (or equivalently, penalize the worst-case distribution of data for including domain i), then we form the robust continuous and discrete optimization problem:

$$\underset{\mathbf{w}\in W}{\text{minimize}} \ \max_{\mathbf{p}\in C} \ \sum_{i=1}^{K} \mathbf{p}_i f_i(\mathbf{w}) - g(\text{supp}\,(\mathbf{p})),$$

with $g : 2^{[K]} \to \mathbb{R}$ a monotone submodular set function. By considering a penalty on the set of nonzero entries of the worst-case distribution, we encode some prioritization of which domains are more or less relevant to us in our application. Then by Theorem 8, we can solve the inner maximization problem (with an appropriate change of signs) by adding a Lagrange multiplier μ and solving a related convex problem.

6.2 General Results

More generally, robust optimization problems can often be expressed as a min-max saddle point optimization problem of a function $q : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{maximize}} \ \min_{\mathbf{y}\in\mathcal{Y}} \ q(\mathbf{x}, \mathbf{y}). \tag{32}$$

This problem is interpreted as maximizing the function $q(\mathbf{x}, \mathbf{y})$ with respect to our available parameters $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, under the worst case choice of additional problem parameters $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^m$ (Ben-Tal et al., 2009).

Given some appropriate structure for the function q, the min-max problem (32) is surprisingly tractable. If we define $Q : \mathcal{X} \to \mathbb{R}$ as:

$$Q(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{x}, \mathbf{y}),$$

we can express the saddle-point problem (32) as:

$$\underset{\mathbf{x} \in \mathcal{X}}{\text{maximize}} \;\; Q(\mathbf{x}). \tag{33}$$

If the function $q(\mathbf{x}, \mathbf{y})$ is concave in $\mathbf{x}$ for any fixed $\mathbf{y} \in \mathcal{Y}$, then the function Q is also concave in $\mathbf{x}$ (Borwein and Lewis, 2006). Moreover, we can compute a subgradient of Q at any $\mathbf{x}_0 \in \mathcal{X}$ as:

$$\nabla_{\mathbf{x}} Q(\mathbf{x}_0) = \nabla_{\mathbf{x}} q(\mathbf{x}_0, \mathbf{y}^*),$$
$$\mathbf{y}^* \in \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmin}} \;\; q(\mathbf{x}_0, \mathbf{y}).$$

In other words, efficiently solving the minimization problem defining Q for an $\mathbf{x}_0 \in \mathcal{X}$ also gives a subgradient of Q. Because Q is concave in $\mathbf{x}$, even a straightforward algorithm such as projected subgradient ascent in the problem (33) will converge to a global optimum.

In this work, we showed that minimization problems in the form of (15) with functions satisfying Assumptions 1-3 can be solved efficiently. Suppose then, that the function $q : \mathcal{X} \times \mathcal{Y}$ is of the form:

$$q(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}, \mathbf{y}) + g(\eta(\mathbf{y}))$$

with $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ concave in $\mathbf{x}$ for any fixed $\mathbf{y}$ and also convex and submodular on $\mathcal{Y} \subseteq \mathbb{R}^n_{\geq 0}$ in $\mathbf{y}$ for any fixed $\mathbf{x}$. If $\eta : \mathcal{Y} \to \mathcal{L}$ satisfies Assumption 3, $g : \mathcal{L} \to \mathbb{R}$ is monotone and submodular, and we assume the set of $\mathbf{y} \in \mathcal{Y}$ such that $\eta(\mathbf{y}) \sqsubseteq \ell$ is a convex subset for any $\ell \in \mathcal{L}$, then the robust optimization problem (32) becomes:

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{maximize}} \;\; \min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) + g(\eta(\mathbf{y})). \tag{34}$$

For a given $\mathbf{x}_0 \in \mathbb{R}^n$, we view the selection of $\mathbf{y} \in \mathcal{Y}$ as a worst-case, or "adversarial" choice of parameters for the function f. The penalty on $\eta(\mathbf{y})$ suggests that the adversarial parameters are selected while considering some preferred structure, such as sparsity. Submodularity here, implies that this adversary pays diminishing prices as it increases the number of parameters it uses.

In addition, Q becomes:

$$Q(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) + g(\eta(\mathbf{y})),$$

which is still the minimum of a family of concave functions, and therefore amenable to subgradient ascent methods as discussed above. A subgradient of Q can easily be computed as:

$$\nabla_{\mathbf{x}} Q(\mathbf{x}_0) = \nabla_{\mathbf{x}} q(\mathbf{x}_0, \mathbf{y}^*) = \nabla_{\mathbf{x}} f(\mathbf{x}_0, \mathbf{y}^*),$$
$$\mathbf{y}^* \in \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmin}} \;\; f(\mathbf{x}_0, \mathbf{y}) + g(\eta(\mathbf{y})).$$

We collect these ideas into the following theorem.

Theorem 10 *Consider the robust optimization problem* (34). *Assume* $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *is concave in* $\mathbf{x} \in \mathcal{X}$ *for any fixed* $\mathbf{y} \in \mathcal{Y}$, *and also convex and submodular in* $\mathbf{y} \in \mathcal{Y}$ *for any fixed* $\mathbf{x} \in \mathcal{X}$. *Let* $\eta : \mathcal{Y} \to \mathcal{L}$ *satisfy Assumption 3,* $g : \mathcal{L} \to \mathbb{R}$ *be a monotone submodular function and assume that for a given* $\ell \in \mathcal{L}$, *the set of* $\mathbf{y} \in \mathcal{Y}$ *such that* $\eta(\mathbf{y}) \sqsubseteq \ell$ *is a convex subset of* $\mathcal{Y}$. *Moreover, let* $\mathcal{Y}$ *be a finitely presentable distributive lattice. For any* $\epsilon \in \mathbb{R}_{>0}$, *let* $T \in \mathbb{Z}_{>0}$ *be of order* $O(\frac{1}{\epsilon^2})$, *meaning as* T *tends to infinity, there exists a constant* $M \in \mathbb{R}_{>0}$ *such that* $T \leq \frac{M}{\epsilon^2}$. *Then* T *iterations of projected subgradient ascent using step lengths* $\eta_i = \frac{1}{\sqrt{T}}$ *produces, in polynomial time, iterates* $\mathbf{x}^{(i)} \in \mathcal{X}$ *for* $i = 1, 2, ..., T$ *such that* $\frac{1}{T}\sum_{i=1}^{T} Q(\mathbf{x}^{(i)}) \leq Q(\mathbf{x}^*) + \epsilon$.

The computational complexity of this approach may be high, as projected subgradient ascent can be slow in practice. However, each sub-problem instance involves a mixed continuous and discrete optimization problem, so this complexity is warranted.

7. Relaxing Submodularity

For the results of Theorem 2 and therefore Corollary 6 and its extensions to apply, Assumptions 1-3 must be met. There are, however, situations where these assumptions may not hold. For example, consider again a quadratic form for $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{p}^T\mathbf{x}, \tag{35}$$

and a monotone and submodular set function $g : 2^{[n]} \to \mathbb{R}$. Then the general lattice optimization problem (P) becomes:

$$\underset{\mathbf{x}\in\mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad \ell(\mathbf{x}) := \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{p}^T\mathbf{x} + g(\text{supp}\,(\mathbf{x})). \tag{36}$$

The assumption that f is submodular on $(\mathbb{R}^n_{\geq 0}, \preceq)$ is equivalent to:

$$\frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \mathbf{Q}_{ij} \leq 0, \quad \text{for all } i \neq j.$$

Moreover, for Corollary 6 to apply, we also need the matrix $\mathbf{Q}$ to be positive semidefinite. These two assumptions are unlikely to both be met by quadratic forms resulting from real data.

Typically, violations of submodularity are handled by suitably relaxing the definition of submodularity with an additive or multiplicative constant (Elenberg et al., 2018; Das and Kempe, 2018). This constant is then propagated through the particular algorithm choice, providing a similarly relaxed optimality guarantee (El Halabi and Jegelka, 2020).

Alternatively, our work focuses on finding exact solutions to these joint problems in an algorithm-agnostic and efficient way. In this spirit, we show in this section how quadratic problems such as (36) can be embedded in another optimization problem satisfying Assumptions 1-3. We then prove conditions under which the solutions to this *lifted* optimization problem–which can be efficiently found, since Assumptions 1-3 are now satisfied–correspond to an exact solution of the original quadratic problem (36).

7.1 Lifting Non-submodular Quadratics

Given the quadratic form for f as in (35), we can decompose the matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ into its submodular and non-submodular parts additively:

$$\mathbf{Q} = \mathbf{Q}^- + \mathbf{Q}^+, \tag{37}$$

$$\mathbf{Q}^-_{ij} = \begin{cases} \mathbf{Q}_{ij}, & i = j \text{ or } \mathbf{Q}_{ij} \leq 0, \\ 0, & \text{otherwise}, \end{cases} \qquad \mathbf{Q}^+_{ij} = \begin{cases} \mathbf{Q}_{ij}, & i \neq j \text{ and } \mathbf{Q}_{ij} > 0 \\ 0, & \text{otherwise}. \end{cases} \tag{38}$$

Then, we define a new, lifted quadratic function $\tilde{f} : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ as:

$$\tilde{f}(\mathbf{z}, \mathbf{w}) = \frac{1}{2} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}^- & \mathbf{Q}^+ \\ \mathbf{Q}^+ & \mathbf{Q}^- \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{q} \\ \mathbf{q} \end{bmatrix}^T \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix}. \tag{39}$$

The lifted function $\tilde{f}$ also has some nice properties that we can use to our advantage.

Lemma 11 *The function $\tilde{f} : \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ defined in (39) is such that for all $(\mathbf{z}, \mathbf{w}) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$:*

$$\tilde{f}(\mathbf{z}, \mathbf{w}) = \tilde{f}(\mathbf{w}, \mathbf{z}), \tag{40}$$

and for all $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$:

$$\tilde{f}(\mathbf{x}, \mathbf{x}) = f(\mathbf{x}). \tag{41}$$

We can similarly lift the function $g : 2^{[n]} \to \mathbb{R}$ to the function $\tilde{g} : 2^{[n]} \times 2^{[n]} \to \mathbb{R}$, defined simply as:

$$\tilde{g}(S, T) = \frac{1}{2} \left(g(S) + g(T) \right). \tag{42}$$

The lifted function $\tilde{g}$ satisfies the same symmetry and embedding properties as the lifted function $\tilde{f}$.

Lemma 12 *The function $\tilde{g}$ defined in (42) is such that for all $(S, T) \in 2^{[n]} \times 2^{[n]}$:*

$$\tilde{g}(S, T) = \tilde{g}(T, S), \tag{43}$$

and for all $A \in 2^{[n]}$:

$$\tilde{g}(A, A) = g(A). \tag{44}$$

With the lifted functions $\tilde{f}$ and $\tilde{g}$ in hand, we define a lifted version of the original quadratic optimization problem (36):

$$\underset{(\mathbf{z}, \mathbf{w}) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad \tilde{\ell}(\mathbf{z}, \mathbf{w}) := \tilde{f}(\mathbf{z}, \mathbf{w}) + \tilde{g}\left(\operatorname{supp}(\mathbf{z}), \operatorname{supp}(\mathbf{w}) \right). \tag{45}$$

If we were to solve this lifted problem and find a solution on the diagonal, i.e., a solution $(\mathbf{z}^*, \mathbf{w}^*)$ such that $\mathbf{z}^* = \mathbf{w}^*$, we immediately recover the solution to the original quadratic problem (36).

Lemma 13 *If the solution to the lifted problem* (45)*, denoted* $(\mathbf{z}^*, \mathbf{w}^*) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$ *is such that* $\mathbf{z}^* = \mathbf{w}^*$*, then the point* $\mathbf{x}^* = \mathbf{z}^* = \mathbf{w}^*$ *is an optimal solution to the original quadratic problem* (36)*.*

Proof By Lemmas 11 and 12, we know that:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) = \ell(\mathbf{z}^*) = \ell(\mathbf{w}^*).$$

Further, by the optimality of $(\mathbf{z}^*, \mathbf{w}^*)$ and by shrinking the feasible set, we have:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) = \ell(\mathbf{z}^*) \leq \min_{\mathbf{z},\mathbf{w} \in \mathbb{R}^n_{\geq 0}} \tilde{\ell}(\mathbf{z}, \mathbf{w}) \leq \min_{\substack{\mathbf{z},\mathbf{w} \in \mathbb{R}^n_{\geq 0} \\ \mathbf{z}=\mathbf{w}}} \tilde{\ell}(\mathbf{z}, \mathbf{w}) = \min_{\mathbf{x} \in \mathbb{R}^n_{\geq 0}} \ell(\mathbf{x}).$$

Therefore, the points $\mathbf{z}^*$ and $\mathbf{w}^*$ are also minimizers of the original problem (36). ■

By Lemma 13, the solution to our initial quadratic problem is embedded in the new lifted problem (45). To use this result, however, we need two key ingredients: the ability to solve the lifted problem exactly and efficiently, and a way to easily produce solutions on the diagonal.

7.2 Efficiently solving the lifted problem

The lifted quadratic problem (45) has a nearly identical form to the original problem (36), but now satisfies Assumptions 1-3, as we prove next. As a result, we can use the approach outlined in Section 4.2 to solve the lifted problem.

To discuss Assumption 1 and submodularity, we define a partial order and lattice on the lifted space $\mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$ so that we can discuss submodularity. In particular, we consider the partial order $\ll$, defined as:

$$(\mathbf{z}, \mathbf{w}) \ll (\mathbf{z}', \mathbf{w}') \qquad \Leftrightarrow \qquad \mathbf{z} \preceq \mathbf{z}' \text{ and } \mathbf{w} \succeq \mathbf{w}', \tag{46}$$

where $\preceq$ denotes the partial order on $\mathbb{R}^n$ previously defined in (5). In words, we order the first part of each pair of vectors in the typical fashion, but reverse the order for the second part. This choice of partial order also defines the join and meet operations:

$$(\mathbf{z}, \mathbf{w}) \veebar (\mathbf{z}', \mathbf{w}') = (\mathbf{z} \curlyvee \mathbf{z}', \mathbf{w} \curlywedge \mathbf{w}') \tag{47}$$

$$(\mathbf{z}, \mathbf{w}) \wedge (\mathbf{z}', \mathbf{w}') = (\mathbf{z} \curlywedge \mathbf{z}', \mathbf{w} \curlyvee \mathbf{w}'), \tag{48}$$

where $\curlyvee$ and $\curlywedge$ are the join and meet operations on $(\mathbb{R}^n, \preceq)$ defined in (6) and (7).

By construction, then, the lifted quadratic function $\tilde{f}$ is submodular on this lattice. Moreover, since it is a quadratic form, simple conditions guarantee its convexity. We pursue convexity here to leverage faster exact algorithms for solving the problem, rather than the more general approach for continuous submodular minimization. Applying the continuous submodular minimization algorithm to this lifted problem while using arbitrarily fine discretization may be of future independent interest.

Lemma 14 *The function* $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^n$ *defined in* (39) *is submodular on the lattice* $(\mathbb{R}^n \times \mathbb{R}^n, \ll)$*. Further,* $\tilde{f}$ *is convex if and only if both* $\mathbf{Q}$ *and* $\mathbf{Q}^+ - \mathbf{Q}^-$ *are positive semidefinite.*

Proof We first note that the lattice $(\mathbb{R}^n \times \mathbb{R}^n, \ll)$ is an *orthant conic lattice*, as defined by Bian et al. (2017). Therefore, by Proposition 2 of Bian et al. (2017), $\tilde{f}$ is submodular on this lattice if and only if:

$$\frac{\partial^2 \tilde{f}}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \leq 0, \tag{49}$$

for all $i, j = 1, 2, ..., n$ or $i, j = n+1, n+2, ..., 2n$ with $i \neq j$ and:

$$\frac{\partial^2 \tilde{f}}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \geq 0, \tag{50}$$

for all $i = 1, 2, ..., n$ and $j = n+1, n+2, ..., 2n$. For our lifted function $\tilde{f}$, its Hessian matrix is exactly:

$$\frac{\partial^2 \tilde{f}}{\partial \mathbf{x}^2} = \begin{bmatrix} \mathbf{Q}^- & \mathbf{Q}^+ \\ \mathbf{Q}^+ & \mathbf{Q}^- \end{bmatrix}.$$

By their construction, the matrices $\mathbf{Q}^+$ and $\mathbf{Q}^-$ satisfy both (49) and (50), and $\tilde{f}$ is submodular on $(\mathbb{R}^n \times \mathbb{R}^n, \ll)$.

For convexity, we note that the Hessian matrix must be positive semidefinite. By the matrix similarity:

$$\frac{1}{2}\begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Q}^+ & \mathbf{Q}^- \\ \mathbf{Q}^- & \mathbf{Q}+ \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^+ - \mathbf{Q}^- & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^+ + \mathbf{Q}^- \end{bmatrix},$$

this holds only when $\mathbf{Q} = \mathbf{Q}^+ + \mathbf{Q}^-$ and $\mathbf{Q}^+ - \mathbf{Q}^-$ are positive semidefinite. ■

Similarly, we define a lattice in the lifted discrete space $2^{[n]} \times 2^{[n]}$ using the partial order $\Subset$ defined as:

$$(S, T) \Subset (S', T') \qquad \Leftrightarrow S \subseteq S' \text{ and } T \supseteq T'.$$

The join and meet operations on $(2^{[n]} \times 2^{[n]}, \Subset)$, denoted by $\Cup$ and $\Cap$ respectively, are:

$$(S, T) \Cup (S', T') = (S \cup S', T \cap T')$$
$$(S, T) \Cap (S', T') = (S \cap S', T \cup T').$$

We can then easily establish that the lifted function $\tilde{g}$ is submodular on the lifted discrete lattice.

Lemma 15 *If the function $g : 2^{[n]} \to \mathbb{R}$ is monotone and submodular, then the lifted function $\tilde{g}$ defined in (42) is submodular on the lattice $(2^{[n]} \times 2^{[n]}, \Subset)$. Moreover, it is monotone and submodular on the product lattice, $(2^{[n]} \times 2^{[n]}, \subseteq)$.*

Proof Take a set $(S, T) \in 2^{[n]} \times 2^{[n]}$ and another set $(S', T') \in 2^{[n]} \times 2^{[n]}$. Then by definition, we have:

$$\begin{aligned} \tilde{g}(S, T) + \tilde{g}(S', T') &= \frac{1}{2}\left(g(S) + g(T) + g(S') + g(T')\right) \\ &\geq \frac{1}{2}\left(g(S \cap S') + g(S \cup S') + g(T \cap T') + g(T \cup T')\right) \\ &= \tilde{g}\left((S, T) \Cup (S', T')\right) + \tilde{g}\left((S, T) \Cap (S', T')\right), \end{aligned}$$

where the inequality follows from the submodularity of g, with $\Cup$ and $\Cap$ the join and meet operations associated with the partial order $\Subset$ on $2^{[n]} \times 2^{[n]}$. By grouping terms differently, we also see that $\tilde{g}$ is also monotone and submodular on the more typical product lattice $(2^{[n]} \times 2^{[n]}, \subseteq)$. ■

Because $\tilde{g}$ is monotone on the product lattice and $\tilde{h}$ is submodular on $(\mathbb{R}^{\mathrm{n}}_{\geq 0} \times \mathbb{R}^{\mathrm{n}}_{\geq 0}, \ll)$, Lemma 5 applies, and we can define the parameterized function $\tilde{h} : 2^{[n]} \times 2^{[n]} \to \mathbb{R}$:

$$\tilde{h}(S,T) = \min_{\substack{\mathbf{z},\mathbf{w} \in \mathbb{R}^n_{\geq 0} \\ \mathrm{supp}(\mathbf{z}) \subseteq S \\ \mathrm{supp}(\mathbf{w}) \subseteq T}} \tilde{f}(\mathbf{z}, \mathbf{w}), \tag{51}$$

and then the solution to:

$$\underset{S,T \in 2^{[n]} \times 2^{[n]}}{\text{minimize}} \quad \tilde{g}(S,T) + \tilde{h}(S,T) \tag{52}$$

corresponds to a solution of the lifted problem (45).

Finally, note that Assumptions 1 and 3 are satisfied by $\tilde{f}$, $\tilde{g}$, the lattices $(2^{[n]} \times 2^{[n]}, \Subset)$ and $(\mathbb{R}^{\mathrm{n}}_{\geq 0} \times \mathbb{R}^{\mathrm{n}}_{\geq 0}, \ll)$, and the mapping $\mathrm{supp} : \mathbb{R}^{\mathrm{n}}_{\geq 0} \times \mathbb{R}^{\mathrm{n}}_{\geq 0} \to 2^{[n]} \times 2^{[n]}$. Therefore, we have the following direct corollary of Theorem 2.

Corollary 16 *The function $\tilde{h} : 2^{[n]} \times 2^{[n]}$ is submodular on the lattice $(2^{[n]} \times 2^{[n]}, \Subset)$.*

Finally, if the non-submodular contribution to the quadratic form is not too large, particularly if $\mathbf{Q}^+ - \mathbf{Q}^-$ is positive semidefinite, then by Lemma 14 $\tilde{f}$ is also convex. Under this assumption, Corollary 6 applies, so we can solve the lifted optimization problem exactly in polynomial time.

Corollary 17 *Under the same assumptions as Corollary 6, if $\mathbf{Q}$ and $\mathbf{Q}^+ - \mathbf{Q}^-$ are both positive semidefinite matrices and $g : 2^{[n]} \to \mathbb{R}$ is monotone and submodular, then the lifted quadratic optimization problem (45) can be solved exactly in polynomial time.*

7.3 Guarantees

Corollary 17 in the previous subsection showed that a quadratic problem that does not satisfy Assumptions 1-3 can be lifted to another quadratic problem that does. Moreover, under mild assumptions on the problem data, the lifted problem can be solved exactly in polynomial time. The question then arises: is this lifted problem's solution useful?

Lemma 13 stated that if we are lucky enough to compute a minimizer to the lifted problem on the diagonal, then it is also necessarily a minimizer of the original quadratic problem. If we are unlucky, however, we would like to still to construct a minimizer of the original problem using the solution we found. The following result shows that this is indeed possible.

Lemma 18 *Let $(\mathbf{z}^*, \mathbf{w}^*) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$ be a solution to the lifted quadratic optimization problem (45). If:*

$$(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \leq 0, \tag{53}$$

then both $(\mathbf{z}^*, \mathbf{z}^*)$ and $(\mathbf{w}^*, \mathbf{w}^*)$ are also minimizers of the lifted problem. By extension, $\mathbf{z}^*$ and $\mathbf{w}^*$ are minimizers of the original quadratic problem (36).

Proof By Proposition 25 (in the appendix), we have that:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) + \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) = 2\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).$$

Re-arranging, and applying the optimality of $(\mathbf{z}^*, \mathbf{w}^*)$, it follows that:

$$(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) = \underbrace{\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) - \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*)}_{\geq 0} + \underbrace{\tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) - \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*)}_{\geq 0} \geq 0.$$

Next, by assumption, $(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \leq 0$, and therefore:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) - \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) + \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) - \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) = 0.$$

If we again re-arrange and apply the optimality of $(\mathbf{z}^*, \mathbf{w}^*)$, we find:

$$0 \leq \tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) - \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) = \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) - \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) \leq 0,$$

and therefore we have:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) = \tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) = \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*),$$

and by Lemma 13 the points $\mathbf{z}^*$ and $\mathbf{w}^*$ are both minimizers of the original quadratic problem (36). ■

Note then that for any minimizer $(\mathbf{z}^*, \mathbf{w}^*)$ of the lifted problem (45), by the submodularity of $\tilde{f}$ and $\tilde{g}$ and the definition of the lattice $(\mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}, \ll)$, we can also construct the minimizer $(\mathbf{z}^* \curlyvee \mathbf{w}^*, \mathbf{z}^* \curlywedge \mathbf{w}^*)$ and its counterpart, $(\mathbf{z}^* \curlywedge \mathbf{w}^*, \mathbf{z}^* \curlyvee \mathbf{w}^*)$. If *any* of these minimizers satisfy the criteria of Lemma 18, then we immediately recover an optimal solution of the original quadratic problem.

The conditions required by Lemma 18 are in fact not only sufficient, but necessary. In particular, any two solutions that are on the diagonal must satisfy them. We defer its proof to the appendix because of its similarity to the proof of Lemma 18.

Lemma 19 *If* $(\mathbf{z}^*, \mathbf{z}^*)$ *and* $(\mathbf{w}^*, \mathbf{w}^*)$ *are minimizers of the lifted problem* (45)*, then:*

$$(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \leq 0.$$

Lemmas 18 and 19 show that the easily verified quadratic form condition on the solutions to the lifted problem are both necessary and sufficient. In practice, we can simply solve the lifted problem and then check if the condition holds.

What might happen if the conditions of Lemma 18 are not satisfied, but we use its suggested minimizer anyways? It turns out that these solutions are still nearly optimal, with the distance from optimality measured using the same necessary and sufficient condition in Lemmas 18 and 19.

Lemma 20 *Let* $\mathbf{x}^* \in \mathbb{R}^n_{\geq 0}$ *be a minimizer of the original quadratic problem* (36)*, and* $(\mathbf{z}^*, \mathbf{w}^*) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0}$ *be a minimizer of the lifted quadratic problem* (45)*. Then:*

$$\min\{\ell(\mathbf{z}^*), \ell(\mathbf{w}^*)\} \leq \ell(\mathbf{x}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).$$

Proof Again applying Proposition 25, we have:

$$\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) + \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) = 2\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).$$

Then, applying the optimality of $(\mathbf{z}^*, \mathbf{w}^*)$, we upper bound the right hand side:

$$\begin{aligned}\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*) + \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*) &= 2\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \\ &\leq 2\tilde{\ell}(\mathbf{x}^*, \mathbf{x}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).\end{aligned}$$

If we divide by two note that the minimum is less than the average, we have:

$$\begin{aligned}&\tilde{\ell}(\mathbf{z}^*, \mathbf{w}^*) \leq 2\tilde{\ell}(\mathbf{x}^*, \mathbf{x}^*) + (\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \\ \Rightarrow &\min\{\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*), \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*)\} \leq \tilde{\ell}(\mathbf{x}^*, \mathbf{x}^*) + \frac{1}{2}(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).\end{aligned}$$

Then, by Lemmas 11 and 12, this implies the result:

$$\min\{\tilde{\ell}(\mathbf{z}^*, \mathbf{z}^*), \tilde{\ell}(\mathbf{w}^*, \mathbf{w}^*)\} = \min\{\ell(\mathbf{z}^*), \ell(\mathbf{w}^*)\} \leq \ell(\mathbf{x}^*) + \frac{1}{2}(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*).$$

■

This series of results suggests the following approach for quadratic problems that violate Assumption 1: lift the problem to a higher-dimensional one satisfying all the required assumptions, solve the new lifted problem, then check if the conditions for Lemma 18 are satisfied. If so, then construct the associated minimizer of the original problem. If the conditions are not satisfied, the value we computed immediately gives an additive bound on the suboptimality of the result.

8. Examples and Computational Evaluation

In this section, we illustrate the proposed theoretical results on several numerical examples involving optimization on the lattices $\mathbb{R}^n_{\geq 0}$ and $2^{[n]}$. We compare against two state-of-the-art techniques: a direct application of the continuous submodular function minimization algorithms outlined by Bach (2019), and the projected subgradient descent method proposed in El Halabi and Jegelka (2020).

The algorithms for continuous submodular function minimization operate by discretizing the domain $\mathbb{R}^n_{\geq 0}$ into k discrete points in each dimension, converting the continuous optimization problem into a submodular minimization problem over a bounded integer lattice. In our examples, we consider the domain $[0, 1]^n \subseteq \mathbb{R}^n_{\geq 0}$ and set the discretization level to $k = 51$ unless otherwise specified. The algorithms for continuous submodular function minimization then solve an equivalent convex optimization problem (defined using a generalized Lovász extension for the integer lattice) using projected subgradient or Frank-Wolfe techniques. In our implementation, we use the Pairwise Frank-Wolfe algorithm to solve this convex problem, with all relevant results plotted in blue and labeled *Cont Submodular*.

The projected subgradient method is known to provide approximation guarantees even in the non-submodular case (El Halabi and Jegelka, 2020), but as shown in Section 4.2,

amounts to a specific choice of algorithms in our theory. The algorithm operates by solving an equivalent convex optimization problem–in particular, minimizing the Lovász extension of $g + H$ over $[0, 1]^n$–using projected subgradient descent. To implement this approach, we use IBM's CPLEX 12.8 constrained quadratic program solver in MATLAB to evaluate the function H (as expressed in (16)) and use Polyak's rule for updating the step size. The relevant results are plotted in red, and labeled *PGD + CPLEX* in figures.

Our approach is agnostic to the choice of convex optimization and submodular set function minimization routines, so we also use CPLEX to evaluate H. To highlight the utility of an algorithm-agnostic approach, we also implement an active-set method for fast non-negative quadratic programming to evaluate H (Bro and De Jong, 1997). For the submodular set function minimization algorithm, we use the minimum-norm point algorithm from Fujishige and Isotani (2011) as implemented in MATLAB by Krause (2010), coupled with the semi-gradient lattice pruning strategy proposed by Iyer et al. (2013) which has quadratic complexity and drastically reduces the problem size. Our results are plotted in black, and labeled *MNP + CPLEX* and *MNP + FNNQP* in figures.

The various methods are given identical cost functions to minimize, and are run until either convergence to suboptimality below 10^{-4} or a maximum of 100 iterations. The experiments were all run on a laptop with an AMD Ryzen 9 4900HS CPU and 16GB of RAM.

8.1 Regularized Sparse Regression

We first examine a regularized sparse regression problem, similar in spirit to (CS). Consider some $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$, $\mathbf{D} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and define the function $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ as:

$$f(\mathbf{x}) = \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2. \tag{54}$$

Then define the monotone submodular set function $g : 2^{[n]} \to \mathbb{R}$ as:

$$g(A) = \begin{cases} \lambda \left[(n-1) + \max(A) - \min(A) + |A| \right], & A \neq \emptyset, \\ 0 & A = \emptyset, \end{cases} \tag{55}$$

with $\lambda \in \mathbb{R}_{\geq 0}$, and $\max(A)$ and $\min(A)$ denoting the largest and smallest index element, respectively, in the set of indices A. This choice of g in the sparse regression problem (P) places a high penalty on large sets of nonzero entries in the vector $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ that are far apart in index.

We generate a series of random problem instances with $m = n$ satisfying the assumption of submodularity on $\mathbb{R}^n_{\geq 0}$ and also the convexity condition of Corollary 6. Let $\text{chol} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ denote a Cholesky decomposition of a positive semidefinite matrix, and construct the matrix $\mathbf{D}$ in (54) as:

$$\mathbf{D} = \text{chol}\left(\frac{1}{2}(\mathbf{C} + \mathbf{C}^T) + n\mathbf{I} \right), \quad \mathbf{C}_{ij} \sim \text{unif}(-1, 0), \text{ for all } i, j = 1, 2, ..., n.$$

This construction guarantees that the function f in (54) is both convex and submodular on $\mathbb{R}^n_{\geq 0}$, satisfying the conditions for Corollary 6. For the parameter $\mathbf{b} \in \mathbb{R}^m$, we use the signal

in the top plot of Figure 1, and we set the regularization strength to $\lambda = 0.05$ so that both the functions f and g play nontrivial roles in the combined objective function.

We plot the results from each algorithm in Figure 1. Because the minimizer of the optimization problem is a representation of $\mathbf{b}$ using structured sparse columns of $\mathbf{D}$, we show the the reconstructed vector $\mathbf{Dx}$ produced by each algorithm in the second, third, and fourth plots of Figure 1. Because there is no reliance on discretization, both the projected subgradient descent and minimum-norm point algorithms produce a much smoother result, as expected.

In the bottom left plot of Figure 1, we show the cost achieved over iterations of each algorithm. The minimum-norm point converges almost immediately to the globally optimal cost, while the projected subgradient descent method takes longer to achieve the same cost. In contrast, the discretization error associated with the continuous submodular function minimization approach prevents it from ever achieving the true optimal cost, by a small amount.

Finally, over a small window of problem sizes, we show the running times of each algorithm in the bottom right plot of Figure 1. Interestingly, our approach presents a compromise between the slow optimality of the projected subgradient descent method and the fast but inexact continuous submodular function minimization algorithm. Moreover, when we take advantage of the extra problem structure to use specialized algorithms, we achieve comparable running times to the continuous submodular minimization algorithm.

8.2 Signal Denoising

We next study a simple denoising example, where we consider a signal $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$, which is corrupted by some additive disturbance $\mathbf{w} \in \mathbb{R}^n$, with $\mathbf{w} \sim \mathcal{N}(0, 0.1\mathbf{I})$. We would like to recover the signal $\mathbf{x}$ from the noisy measurements $\mathbf{y} = \mathbf{x} + \mathbf{w}$, under the assumption that the true signal $\mathbf{x}$ is smooth (meaning variations between adjacent entries ought to be small), and that the meaningful content arrived in a small number of contiguous sets of entries.

We can express the desire to match the noisy signal $\mathbf{y}$ with a smooth one with the convex and submodular function $f : \mathbb{R}^n \to \mathbb{R}$ defined as:

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\| + \mu \sum_{i=1}^{n-1} (\mathbf{x}_i - \mathbf{x}_{i+1})^2 . \tag{56}$$

The first term promotes matching the slightly corrupted signal, while the quadratic penalty on adjacent entries of $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ promotes smoothness.

Similarly, we can express the knowledge of a small and contiguous set of nonzero entries in the vector $\mathbf{x}$ with the monotone submodular set function $g : 2^{[n]} \to \mathbb{R}$ defined by:

$$g(A) = \lambda \left(|A| + \#\text{int}(A)\right), \tag{57}$$

where $\lambda \in \mathbb{R}_{\geq 0}$, and the function $\#\text{int}(A)$ counts the number of sets of contiguous indices in the set A. This set function is smallest on subsets with a small number of entries that are adjacent in index.

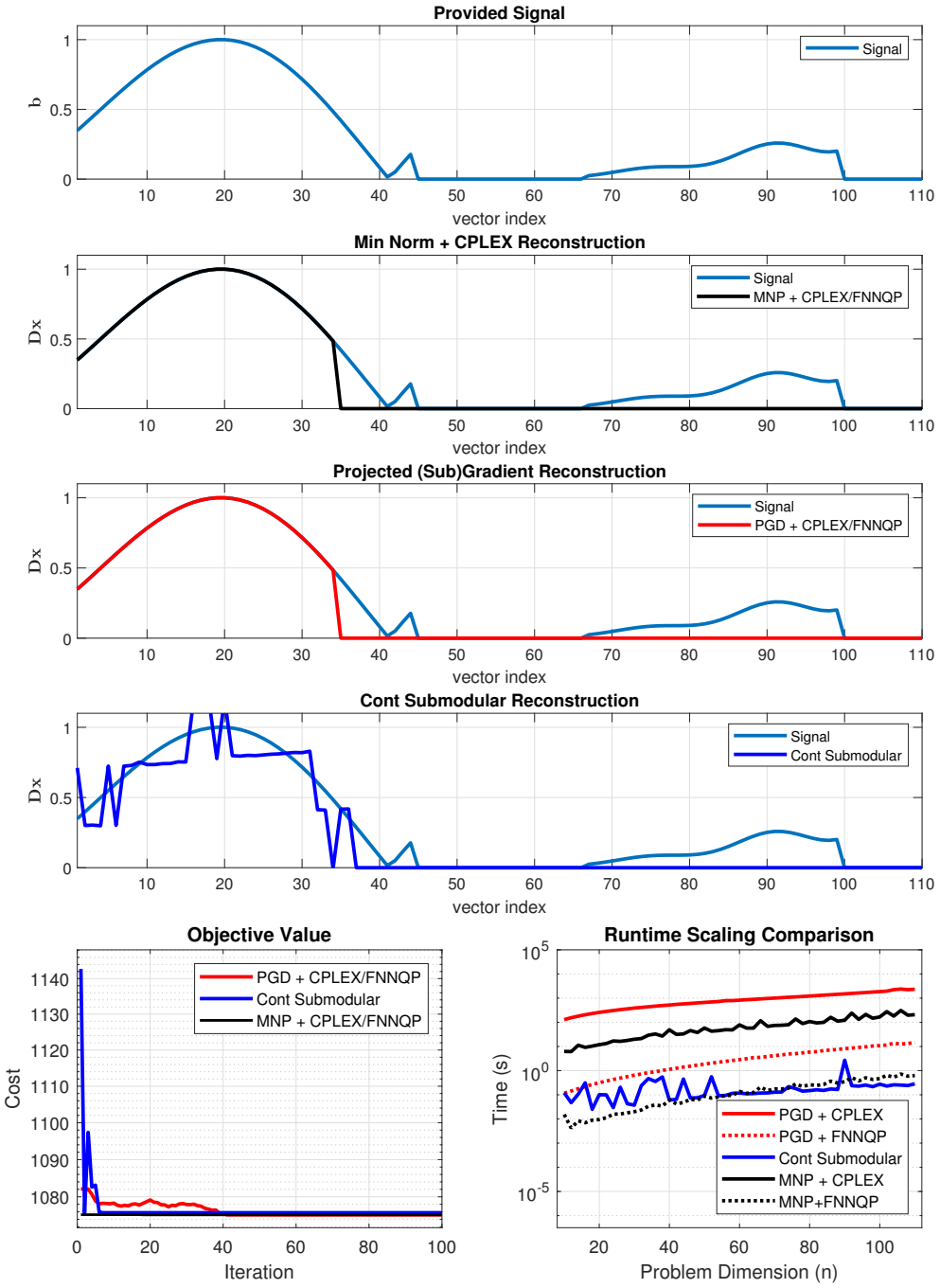

Figure 1: Results from the sparse regression problem simulations. The reconstructed signal representations using columns of $\mathbf{D}$ created by each algorithm are shown in the second, third, and fourth plot. Note the solutions produced by projected subgradient and the minimum-norm point algorithm are identical. We plot the cost function value over each algorithm's iterations in the bottom left, while in the bottom right we compare the running times of the algorithms over a small window of problem dimensions.

For experiments, we use the signal $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ shown in the top plot of Figure 2, with the noise-corrupted measurements $\mathbf{x} + \mathbf{w} = \mathbf{y} \in \mathbb{R}^n$ with an example shown in dotted orange. We then let $\mu = 0.8$ in (56) and $\lambda = 0.05$ in (57) so that the overall problem's cost function has nontrivial contributions from both the smoothness-promoting function and the sparsity-inducing regularizer. In this case, for the continuous submodular algorithm we discretize the compact set $[0, 1]^n \subseteq \mathbb{R}^n$ into $k = 51$ distinct values per index.

We show the resulting denoised signals in the second, third, and fourth plots in Figure 2, with the running time comparison over a small window of problem dimensions in the bottom right. The discretization of the domain in the continuous submodular function minimization approach produces artifacts in the reconstructed signal, whereas the result of the projected subgradient and minimum-norm point algorithms are smoother with smaller sets of nonzero entries. We see once more that our proposed minimum-norm point algorithm poses a compromise between speed and accuracy, providing guaranteed global optimality without the high running time of projected subgradient descent. Moreover, when we use more specialized algorithms for each sub-problem, we achieve competitive performance with the continuous submodular minimization algorithm.

We also compare the objective value achieved during the iterations of each algorithm for a single instance in the bottom left plot of Figure 2 with $n = 100$. Again, the minimum-norm point algorithm converges almost immediately to the minimum alongside the projected subgradient method, while the continuous submodular function minimization approach's discretization error prevents it from achieving full global optimality.

8.3 Price optimization with start-up costs

In price optimization problems, we are asked to determine prices for a set of products that maximizes the expected profit while considering any inter-product demand effects caused by these prices (Ito and Fujimaki, 2016, 2017). Usually this process relies on a simple predictive model for the relationship between the price of an item and its demand, which we can easily derive with a regression technique. Given a predictive model of the pricing-demand relationship and a characterization of our cost for each product, we want to determine the optimal pricing strategy that maximizes our profit.

Let $\mathbf{c}_i \in \mathbb{R}_{\geq 0}$ and $\mathbf{p}_i \in \mathbb{R}_{\geq 0}$ denote the cost and retail price per unit, respectively, of each item of each item $i = 1, 2, ..., n$. Let the function $d : \mathbb{R}^{\mathrm{n}}_{\geq 0} \to \mathbb{R}^{\mathrm{n}}_{\geq 0}$ be the predictive demand model, meaning that given a set of prices $\mathbf{p}$ it estimates the number of sales (or demand) of the products. The estimated total profit of a pricing $\mathbf{p}$ can then be described by the function:

$$f(\mathbf{p}) = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{c}_i) d(\mathbf{p})_i. \tag{58}$$

Without loss of generality, we assume there is a minimum loss we are willing to accept for each item, meaning there is a lower bound $\underline{\mathbf{p}} \in \mathbb{R}^{\mathrm{n}}_{\geq 0}$, and that if $\mathbf{p}_i = \underline{\mathbf{p}}_i$, we will not sell product i.

While the expression for profit (58) includes the cost of each item, it does not account for any start-up costs associated with providing them. In particular, to provide an item, we

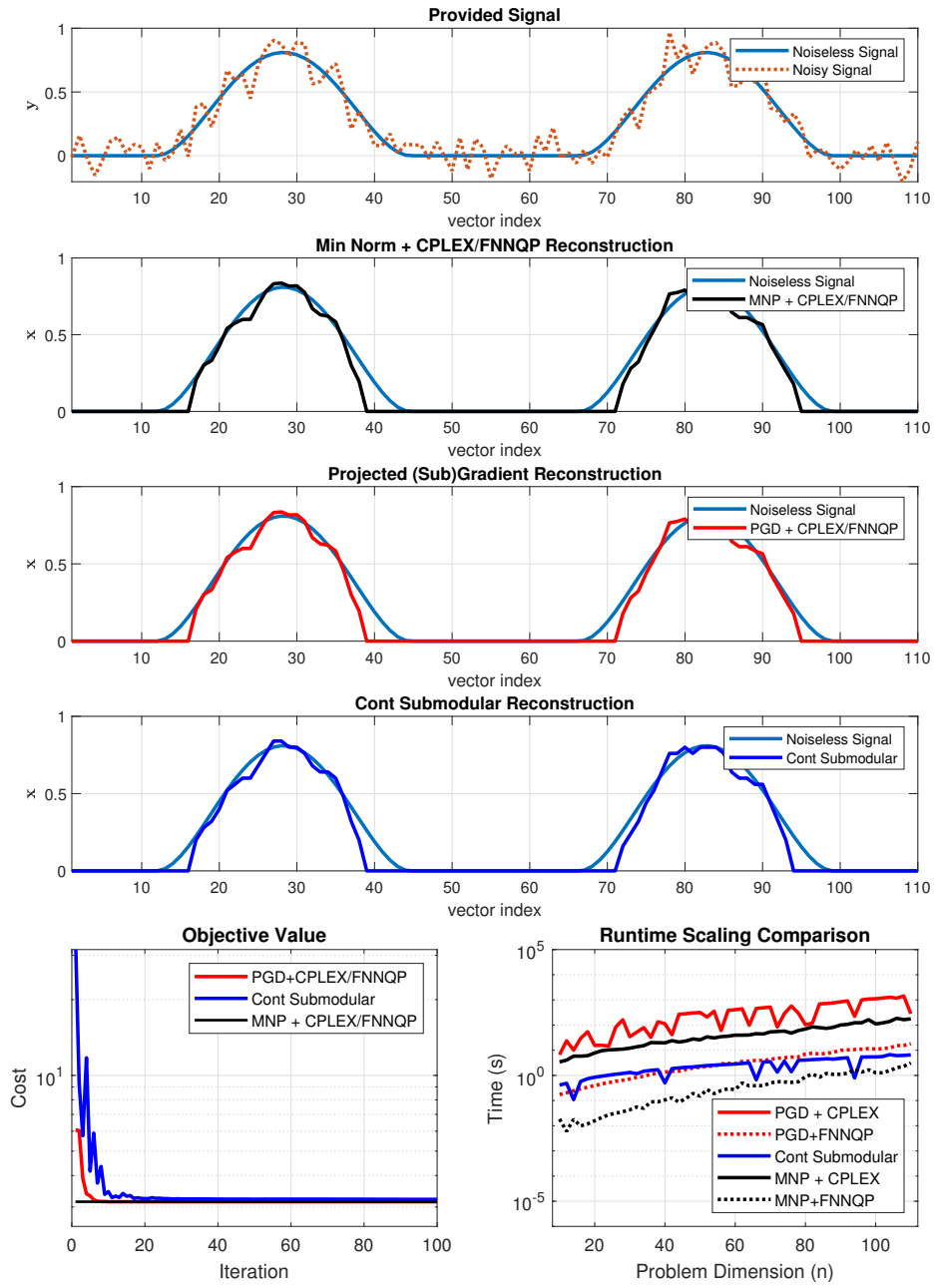

Figure 2: Results of the denoising problem simulations. The true signal and its noisy counterpart are shown in the top plot. The second, third, and fourth plots show the denoised signals produced by each of the three algorithms. Note that the results from the minimum-norm point algorithm and the projected subgradient descent method are identical. The bottom left plot shows the objective value across iterations for $n = 100$, and bottom right shows the running times of each algorithm for a window of problem dimensions.

may have to order it from a supplier and have it shipped to our facilities, paying various logistical fees to do so. We pay these fees regardless of the *quantity* of products, meaning they are a function purely of which items we choose to stock. Moreover, in many cases these logistical costs are lumped together between items, such as when sourcing multiple products from the same supplier.

More mathematically, assume we have $k \in \mathbb{Z}_{>0}$ groups of products with shared start-up costs, with each group represented as a subset $G_i \subseteq [n]$, each with some start-up cost $\mathbf{w}_i$. Then the total incurred start-up costs of a subset of provided products S can be expressed with a set function $g : 2^{[n]} \to \mathbb{R}$:

$$g(S) = \sum_{\substack{k \in [n] \\ S \cap G_k \neq \emptyset}} \mathbf{w}_i. \tag{59}$$

We apply this set function to the set of products we choose to sell, $\operatorname{supp}\left(\mathbf{p} - \underline{\mathbf{p}}\right) \subseteq [n]$. In this work, without loss of generality we let $\underline{\mathbf{p}} = 0$, which implies that an item priced at $\mathbf{p}_i = \underline{\mathbf{p}}_i$ earns no reward and also has no impact on the demand of the other products. By carefully defining the demand model d and costs c, we can enforce this property for any desired minimum price $\underline{\mathbf{p}}$.

The true underlying demand model d is unknown in practice. In a small time window, however, we can use historical data to build a local linear approximation for it, $\hat{d} : \mathbb{R}^{\mathrm{n}}_{\geq 0} \to \mathbb{R}^{\mathrm{n}}_{\geq 0}$:

$$\hat{d}(\mathbf{p}) = \boldsymbol{\beta}\, \mathbf{p} + \boldsymbol{\alpha},$$

with $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$. The entries $\boldsymbol{\beta}_{ij}$ describe the impact that the price of product i has on the demand for product j, sometimes referred to as the *elasticity of demands*(Ito and Fujimaki, 2016, 2017). Using this model, the estimated expected profit (58) is a quadratic function:

$$f(\mathbf{p}) = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{c}_i)\hat{d}(\mathbf{p})_i = \mathbf{p}^T \boldsymbol{\beta}\, \mathbf{p} + \mathbf{p}^T(\boldsymbol{\alpha} - \boldsymbol{\beta}^T \mathbf{c}) - \mathbf{c}^T \boldsymbol{\alpha}\,.$$

Combining the expected profits with the start-up costs, we are faced with the optimization problem:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & -\mathbf{p}^T \boldsymbol{\beta}\, \mathbf{p} - \mathbf{p}^T(\boldsymbol{\alpha} - \boldsymbol{\beta}^T \mathbf{c}) + \mathbf{c}^T \boldsymbol{\alpha} + g\left(\operatorname{supp}\left(\mathbf{p} - \underline{\mathbf{p}}\right)\right) \\ \text{subject to} & \mathbf{p} \geq \underline{\mathbf{p}}. \end{array} \tag{60}$$

We create this scenario with real retail sales data collected from a UK-based online retail store available in the UCI Machine Learning Repository (Dua and Graff, 2017; Chen et al., 2012). We use this data to estimate the matrix $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$ and vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ with simple ridge regression. To make the pricing problem (60) well-posed, we also enforce a weak diagonal dominance constraint on $\boldsymbol{\beta}$. In addition to making the problem well-posed, this constraint enforces the intuition that the most relevant factor in each product's demand is its own prices.

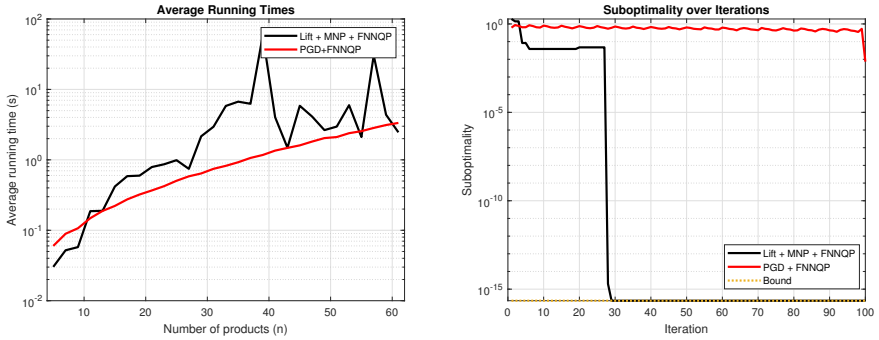

Figure 3: Results of the price optimization problem simulations. We show the running times of each algorithm for various problem sizes (left) and the achieved cost across iterations of the algorithms for a problem of size $n = 20$ (right). The dotted line below indicates the guaranteed lower bound on the optimal solution provided by our lift.

Even with a diagonal dominance constraint, the cross-terms $\boldsymbol{\beta}_{ij}$ with $i \neq j$ can easily be either positive or negative, depending on the demand and price relationships of the products. As a result, we cannot directly apply our parameterization method. We can, however, use the quadratic structure of (60) and follow the results of Section 7 to lift the pricing problem into a new quadratic problem amenable to our parameterization approach.

We compare our parameterization approach to solving (60) against the projected subgradient descent method applied directly to the original quadratic program for 100 iterations. This algorithm gives near-optimality guarantees, but explicitly computing the associated bound is NP-Hard. Alternatively, our quadratic lifting approach gives an easily computable additive suboptimality guarantee in Lemma 20 at the cost of solving a larger problem instance. This trade-off is highlighted in the plot of running times across varying problem sizes and the achieved cost across over iterations of each algorithm for an instance of $n = 20$ in Fig. 3.

We could also, in principle, use the continuous submodular minimization algorithm to solve the lifted quadratic problem. However, this approach will still suffer inaccuracy from the discretization step, and further, runs slower than the other algorithms that take advantage of the quadratic problem structure.

8.4 Discretization Error Dependence

In this section, we explore the relationship between the continuous submodular function minimization algorithm's discretization error and its running time. To this end, we ran instances of the sparse regression example with the modified range function penalty, using a discretization resolution in each dimension ranging from $k = 50$ to $k = 400$.

The minimum cost achieved at each discretization level k is shown in the left plot of Figure 4. Similarly, the associated running times of the algorithm are shown in the right-hand plot of Figure 4. Interestingly, near the value of $k = 250$, the achieved cost becomes effectively optimal, but the running time increases by an order of magnitude.

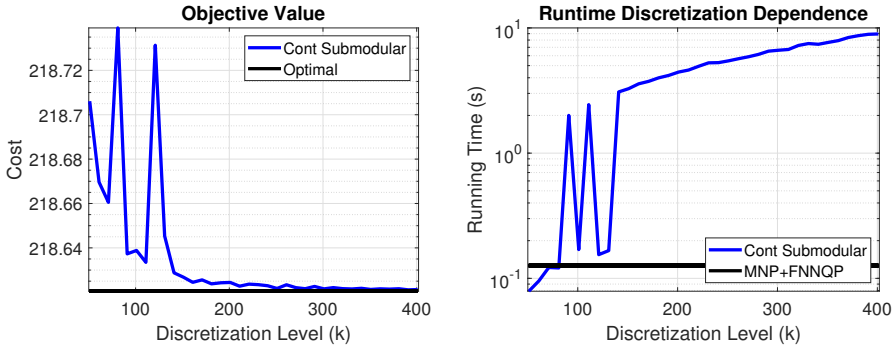

Figure 4: Results highlighting the role of the discretization resolution k on the continuous submodular algorithm's optimality (left) and running times (right) in an instance of the sparse regression problem with $n = 100$.

To give a coarse estimate on the origin of higher running times for projected subgradient descent and the minimum-norm point algorithms, we note that the computational cost of each iteration is dominated by the cost of computing the Lovàsz extension of H. This computation has time complexity $O(n \log n + nEO)$, where EO is the complexity of evaluating H. If H is evaluated through convex optimization, many generic interior-point methods have time complexity that is approximately $EO = O(n^3)$. Therefore, each iteration of the minimum-norm point algorithm and the projected subgradient descent algorithm might have complexity on the order of $O(n \log n + n^4)$. When using the fast non-negative quadratic programming algorithm, however, each evaluation operation is typically much lower than the generic $O(n^3)$. Moreover, the lattice reduction technique of Iyer et al. (2013) runs in approximately $O(n^2)$, and reduces the problem size drastically in many problems, as seen above.

9. Conclusions

In this work, we showed that model-fitting problems with structure-promoting regularizers could be expressed as optimization problems defined over two connected lattices. Using submodularity theory, we derived conditions on these functions and their domains under which we can directly solve these problems exactly and efficiently. We focused on continuous and Boolean lattices, and derived conditions under which an agnostic combination of submodular set function minimization and convex optimization algorithms can compute the exact solution in polynomial time.

We then extended this theory to handle optimization problems with simple continuous or discrete budget constraints on the model parameters. We did this by naively adding the constraint to the cost with a Lagrange multiplier, but then used submodular function theory to solve for all possible Lagrange multiplier values with a single convex optimization problem. We also highlighted robust or adversarial optimization scenarios, where our exact solutions could provide subgradients to be used in globally convergent ascent methods.

Finally, we acknowledged there may be scenarios where our sufficient conditions are violated, and sought a way to weaken them without sacrificing our algorithm-agnostic approach.

To do so, we identified a class of quadratic programming problems that can be lifted to problems satisfying our conditions. We then proved that the solutions of the lifted problem– which can then be found in polynomial time using our previously developed techniques– give provably optimal or near-optimal solutions to the original problem. Moreover, the additive approximation bound we provide is simple to compute, unlike existing guarantees in literature that involve constants that are NP-Hard to compute.

Acknowledgments

We would like to acknowledge the support for this work provided by the US Army Research Laboratory (ARL) under Cooperative Agreement W911NF-17-2-0196.

Appendix A. Submodularity, Lattice Morphisms, and Least Squares

There is a massive body of work that identifies conditions under which compressed sensing problems of the form:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + |\text{supp}\,(\mathbf{x})\,|, \tag{61}$$

for $\mathbf{A}\in\mathbb{R}^{m\times n}$ (with normalized unit norm columns, without loss of generality) and $\mathbf{b}\in\mathbb{R}^m$ can be efficiently solved by a convex relaxation of the ℓ_0 pseudo-norm to the ℓ_1 norm:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \|\mathbf{x}\|_1,$$

with $\|\mathbf{x}\|_1=\sum_{i=1}^n|\mathbf{x}_i|$. The majority of these conditions rely on the matrix $\mathbf{A}$ being "close to an isometry", or "nearly orthogonal". In this appendix, we highlight how these near-orthognality conditions on the matrix $\mathbf{A}$ can be related to the assumptions made in this work.

Interestingly, any least-squares problem in the form of (61) can be written as a least-squares problem over $\mathbb{R}^n_{\geq 0}$, by considering auxiliary variables:

$$\mathbf{x}=\mathbf{x}^+-\mathbf{x}^-, \quad \mathbf{x}^+,\mathbf{x}^-\in\mathbb{R}^n_{\geq 0}.$$

Using these new variables, the least squares problem (61) becomes:

$$\underset{\mathbf{x}^+,\mathbf{x}^-\in\mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad \left\|\begin{bmatrix}\mathbf{A} & -\mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}^+\\ \mathbf{x}^-\end{bmatrix}-\mathbf{b}\right\|_2^2 + |\text{supp}\left(\mathbf{x}^+-\mathbf{x}^-\right)|.$$

If we assume (without loss of generality) that at most one of $\mathbf{x}_i^+$ or $\mathbf{x}_i^-$ are nonzero for each $i=1,2,...,n$, then we can equivalently write:

$$\underset{\mathbf{x}^+,\mathbf{x}^-\in\mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad \begin{bmatrix}\mathbf{x}^+\\ \mathbf{x}^-\end{bmatrix}^T\begin{bmatrix}\mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{A}\\ -\mathbf{A}^T\mathbf{A} & \mathbf{A}^T\mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}^+\\ \mathbf{x}^-\end{bmatrix} - 2\mathbf{b}^T\begin{bmatrix}\mathbf{A} & -\mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}^+\\ \mathbf{x}^-\end{bmatrix}$$
$$+|\text{supp}\left(\mathbf{x}^+\right)|+|\text{supp}\left(\mathbf{x}^-\right)|.$$

In this lifted problem, Assumption 1 states that the cost function must be submodular on $\mathbb{R}^n_{\geq 0}\times\mathbb{R}^n_{\geq 0}$. For our lifted problem's cost function, this assumption is equivalent to the condition:

$$\begin{aligned}\left(\mathbf{A}^T\mathbf{A}\right)_{ij}&\leq 0, \quad \text{for all } i\neq j\\ -\left(\mathbf{A}^T\mathbf{A}\right)_{ij}&\leq 0, \quad \text{for all } i,j.\end{aligned}$$

This set of conditions in turn implies that $\left(\mathbf{A}^T\mathbf{A}\right)_{ii}\geq 0$ for all i, which is always satisfied, but also that $\left(\mathbf{A}^T\mathbf{A}\right)_{ij}=0$ for all $i\neq j$.

By this analysis, any arbitrary least-squares problem with a monotone subset penalty can be converted to a nonnegative least-squares problem satisfying Assumptions 1-3 and the

required convexity for Theorem 2 if $\mathbf{A}$ is orthogonal. The nearness of the matrix $\mathbf{A}$ to satisfying this condition is often measured with the notion of its *coherence*:

$$\max_{i \neq j} \left(\mathbf{A}^T \mathbf{A}\right)_{ij},$$

which is commonly used to identify well-structured instances of least-squares problems (Rauhut, 2010).

Appendix B. Continuous Budget Constraints

In this appendix, we prove the relevant results for continuous budget constraints. We let $f_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ and $W_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be continuous functions such that $f_i(0) = W_i(0) = 0$ for all $i = 1, 2, ..., n$. We further assume that each W_i is strictly increasing for each i. Then define the function $H_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\leq 0}$:

$$H_i(\alpha) = \min_{\mathbf{z} \geq 0} f_i(\mathbf{z}) + \alpha W_i(\mathbf{z}). \tag{62}$$

We first note that H_i is monotone in α.

Proposition 21 *The function $H_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\leq 0}$ is monotone in α for all $i = 1, 2, ..., n$. It is strictly increasing for all $\alpha \in [0, c]$, where $c \in \mathbb{R}_{\geq 0}$ is the smallest constant such that $H_i(c) = 0$. Additionally, H_i is constant and zero on the interval $[c, \infty[$.*

Proof Consider $\alpha, \beta \in \mathbb{R}_{\geq 0}$, with $\alpha \leq \beta$, and define the points $\mathbf{z}^\alpha \in \mathbb{R}_{\geq 0}$ and $\mathbf{z}^\beta \in \mathbb{R}_{\geq 0}$ as:

$$\mathbf{z}^\alpha \in \operatorname*{argmin}_{\mathbf{z} \geq 0} f_i(\mathbf{z}) + \alpha W_i(\mathbf{z}),$$
$$\mathbf{z}^\beta \in \operatorname*{argmin}_{\mathbf{z} \geq 0} f_i(\mathbf{z}) + \beta W_i(\mathbf{z}).$$

Note that for any $\alpha \in \mathbb{R}_{\geq 0}$, because $\mathbf{z} = 0$ is a feasible point in the minimization defined in (62):

$$\begin{aligned} H_i(\alpha) &= \min_{\mathbf{z} \geq 0} f_i(\mathbf{z}) + \alpha W_i(\mathbf{z}) \\ &\leq f_i(0) + \alpha W_i(0) = 0, \end{aligned}$$

thus H_i is bounded above by zero. Moreover, observe that by optimality of $\mathbf{z}^\alpha$:

$$H_i(\alpha) = f_i(\mathbf{z}^\alpha) + \alpha W_i(\mathbf{z}^\alpha) \leq f_i(\mathbf{z}) + \alpha W_i(\mathbf{z}), \quad \text{for all } \mathbf{z} \geq 0.$$

Moreover, because $W_i(0) = 0$ and W_i is increasing, $W_i(\mathbf{z}) \geq 0$. Then, because $\alpha \leq \beta$:

$$\begin{aligned} H_i(\alpha) &= f_i(\mathbf{z}^\alpha) + \alpha W_i(\mathbf{z}^\alpha) \\ &\leq f_i(\mathbf{z}) + \alpha W_i(\mathbf{z}) \\ &\leq f_i(\mathbf{z}) + \beta W_i(\mathbf{z}), \quad \text{for all } \mathbf{z} \geq 0. \end{aligned}$$

This inequality is strict when $\alpha < \beta$ and $W_i(\mathbf{z}^\alpha) \neq 0$, or equivalently $H_i(\alpha) < 0$. In particular, because $\mathbf{z}^\beta \geq 0$:

$$H_i(\alpha) \leq f_i(\mathbf{z}^\beta) + \beta W_i(\mathbf{z}^\beta) = H_i(\beta),$$

with strict inequality when $H_i(\alpha) < 0$. Therefore H_i is monotone and strictly increasing for all $\alpha \in \mathbb{R}_{\geq 0}$ such that $H_i(\alpha) < 0$. Because it is also bounded above by zero, monotonicity implies that once $H_i(c) = 0$ for some $c \in \mathbb{R}_{\geq 0}$, it is zero for all $\beta \geq c$. ■

Let $g : 2^{[n]} \to \mathbb{R}$ be a monotone submodular set function, and consider a family of optimization problems parameterized by $\mu \in \mathbb{R}_{\geq 0}$:

$$\underset{A \in 2^{[n]}}{\text{minimize}} \;\; g(A) + \sum_{i \in A} H_i(\mu). \tag{63}$$

Given Proposition 21, we know that $H_i(0) \leq 0$ for all $i = 1, 2, ..., n$. If there exists an $i \in [n]$ such that $H_i(0) = 0$, Proposition 21 further states that $H_i(\alpha)$ is also zero for all $\alpha \geq 0$. Moreover, because g is monotone, we know:

$$\begin{aligned} g(A) + \sum_{i \in A} H_i(\alpha) &= g(A) + \sum_{i \in A \setminus \{j\}} H_i(\alpha) \\ &\geq g(A \setminus \{j\}) + \sum_{i \in A \setminus \{j\}} H_i(\alpha). \end{aligned}$$

In words, because g is monotone and $H_i(\alpha)$ is zero for all α, we can always reduce the cost of a subset by removing i. Equivalently, we can simply remove i from the ground set of elements.

We then follow the analysis in Bach (2013), generalizing as needed to accommodate for the non-strict monotonicity of H_i.

Proposition 22 *(Proposition 8.2 in Bach 2013) Let A^α and A^β be minimal (i.e., smallest in size) minimizers for* (63) *with respective parameters α and β, with $\alpha < \beta$. Then $A^\beta \subseteq A^\alpha$.*

Proof By the optimality of A^α and A^β, we have:

$$g(A^\alpha) + \sum_{i \in A^\alpha} H_i(\alpha) \leq g(A^\alpha \cup A^\beta) + \sum_{i \in A^\alpha \cup A^\beta} H_i(\alpha) \tag{64}$$

$$g(A^\beta) + \sum_{i \in A^\beta} H_i(\beta) \leq g(A^\alpha \cap A^\beta) + \sum_{i \in A^\alpha \cap A^\beta} H_i(\beta). \tag{65}$$

If we sum these inequalities and apply the submodularity of g, we have:

$$\begin{aligned} g(A^\alpha \cup A^\beta) + g(A^\alpha \cap A^\beta) + &\sum_{i \in A^\alpha \cup A^\beta} H_i(\alpha) + \sum_{i \in A^\alpha \cap A^\beta} H_i(\beta) \\ &\geq g(A^\alpha) + g(A^\beta) + \sum_{i \in A^\alpha} H_i(\alpha) + \sum_{i \in A^\beta} H_i(\beta) \\ &\geq g(A^\alpha \cup A^\beta) + g(A^\alpha \cap A^\beta) + \sum_{i \in A^\alpha} H_i(\alpha) + \sum_{i \in A^\beta} H_i(\beta). \end{aligned} \tag{66}$$

Subtracting equations (64) and (65) from (66), we have:

$$\sum_{i \in A^\alpha \cup A^\beta} H_i(\alpha) + \sum_{i \in A^\alpha \cap A^\beta} H_i(\beta) \geq \sum_{i \in A^\alpha} H_i(\alpha) + \sum_{i \in A^\beta} H_i(\beta)$$
$$\Rightarrow \sum_{i \in A^\beta \setminus A^\alpha} [H_i(\beta) - H_i(\alpha)] \leq 0. \tag{67}$$

By Proposition 21, as $\alpha < \beta$, each $H_i(\beta) - H_i(\alpha)$ in the summation (67) is strictly positive, or $H_i(\alpha) = H_i(\beta) = 0$. But if $H_i(\alpha) = H_i(\beta) = 0$, as g is monotone, we may remove i from both A^α and A^β and decrease the cost in (63), contradicting the minimality of A^α and A^β.

By this argument, the left-hand side of inequality (67) is the sum of strictly positive terms. However, it is bounded above by zero, so it must therefore be the empty summation, i.e., $A^\beta \setminus A^\alpha = \emptyset$, and therefore $A^\beta \subseteq A^\alpha$. ■

We now identify a related convex optimization problem:

$$\underset{\mathbf{u} \in \mathbb{R}^n_{\geq 0}}{\text{minimize}} \;\; g_L(\mathbf{u}) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon + \mathbf{u}_i} H_i(\alpha) d\alpha. \tag{68}$$

A classical result in submodular function theory establishes that the Lovàsz extension g_L is convex if and only if g is submodular (Lovász, 1983). Moreover, $\int_{\epsilon}^{\epsilon+\mathbf{u}_i} H_i(\alpha)d\alpha$ is convex if and only if H_i is monotone in α, which is true by Proposition 21. Therefore, problem (68) is a convex optimization problem.

We now establish a relationship between the parameterized family of set function minimization problems (63) and the convex optimization problem (68).

Proposition 23 *(Proposition 8.3 in Bach 2013) Given the (minimal) solutions A^α to the set function minimization problem* (63) *for all values of the parameter $\alpha \geq \epsilon$, define the vector $\mathbf{u}^* \in \mathbb{R}^n_{\geq 0}$ defined by:*

$$\mathbf{u}^*_i = \sup\left(\{\alpha \in \mathbb{R}_{\geq 0} \mid i \in A^\alpha\}\right).$$

Then the vector $\mathbf{u}^*$ is the minimizer of the convex optimization problem (68)*.*

Proof For $\alpha \geq 0$ small enough (as, without loss of generality, $H_i(0) < 0$ for all i), we have $H_i(\alpha) < 0$ for all $i = 1, 2, ..., n$. Because g is monotone, for this α, the optimal A^α is equal to $\{1, 2, ..., n\}$, and thus $\mathbf{u}$ is well defined for all $i = 1, 2, ..., n$.

For simplicity, we use the notation $\{\mathbf{u} \geq \mu\}$ to denote the set:

$$\{\mathbf{u} \geq \mu\} = \{i \in \{1, 2, ..., n\} \mid \mathbf{u}_i \geq \mu\},$$

for any $\mathbf{u} \in \mathbb{R}^n$ and $\mu \in \mathbb{R}$. Then for any $\mu \geq 0$, we have:

$$\begin{aligned} g_L(\mathbf{u}) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon+\mathbf{u}_i} H_i(\mu)d\mu &= g_L(\mathbf{u}+\mathbf{1}\epsilon) - \epsilon g(\{1,2,..,n\}) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon+\mathbf{u}_i} H_i(\alpha)d\alpha \\ &= \int_{0}^{\infty} g(\{\mathbf{u}+\mathbf{1}\epsilon \geq \mu\})d\mu + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon+\mathbf{u}_i} H_i(\alpha)d\alpha - \epsilon g(\{1,2,...,n\}) \\ &= \int_{\epsilon}^{\infty} \left[g(\{\mathbf{u}+\mathbf{1}\epsilon \geq \mu\}) + \sum_{i=1}^{n} \mathbb{1}_{\{\mathbf{u_i}+\epsilon \geq \mu\}} H_i(\mu) \right] d\mu, \end{aligned} \tag{69}$$

where we used the indicator function defined as:

$$\mathbb{1}_{\{\mathbf{u_i}^*+\epsilon \geq \mu\}} = \begin{cases} 1, & \mathbf{u}_i^* + \epsilon \geq \mu \\ 0, & \text{otherwise.} \end{cases}$$

In the right-hand side of (69), every $\mu \geq \epsilon$ in the integral defines a set function minimization for which the optimal subset is A^μ. Because we constructed $\mathbf{u}^*$ as the minimizer to each of these optimal subsets, the value at $\mathbf{u}^*$ must be lower than all other $\mathbf{u}$, leading to the inequality:

$$\begin{aligned} g_L(\mathbf{u}^*) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon+\mathbf{u}_i^*} H_i(\mu)d\mu &\leq \int_{\epsilon}^{\infty} \left[g(\{\mathbf{u}+\mathbf{1}\epsilon \geq \mu\}) + \sum_{i=1}^{n} \mathbb{1}_{\{\mathbf{u_j}+\epsilon \geq \mu\}} H_j(\mu) \right] d\mu \\ &= g_L(\mathbf{u}) + \sum_{i=1}^{n} \int_{\epsilon}^{\epsilon+\mathbf{u}_i^*} H_i(\mu)d\mu, \end{aligned}$$

for all other $\mathbf{u} \in \mathbb{R}^n_{\geq 0}$, and therefore $\mathbf{u}^*$ is optimal for (68). ■

Proposition 23 establishes the relationship between the parameterized family of optimization problems (63) and the convex optimization problem (68). We state the next theorem without proof, as it requires no special modifications for our conditions.

Proposition 24 *(Proposition 8.4 in Bach 2013) If $\mathbf{u}^*$ is the minimizer for the convex optimization problem* (68)*, then for all $\mu \geq \epsilon$, the minimal minimizer of the corresponding set function minimization in* (63) *is:*

$$A^\mu = \{i \in \{1,2,...,n\} \mid \mathbf{u}_i^* > \mu\}.$$

This sequence of propositions ultimately abuses the interpretation of the Lovàsz extension as an integral, and states that optimizing over the integral itself (the convex problem) and optimizing over the integrated functions for all integration variables (the set functions) is equivalent.

A noteworthy addendum is that in the definition of H_i, we could equivalently perform scalar minimization over a closed subset of $\mathbb{R}_{\geq 0}$, and the analysis would still follow through. This alteration would result in effectively "capping" the H_i functions from below, which retains the monotonicity properties necessary for the proofs.

Appendix C. A useful symmetry property

The lifted quadratic cost function $\tilde{c} : \mathbb{R}^{\mathrm{n}}_{\geq 0} \times \mathbb{R}^{\mathrm{n}}_{\geq 0} \to \mathbb{R}$ satisfies a convenient property that we abuse to prove several results. We prove it here.

Proposition 25 *Let $\tilde{\ell}$ be defined as in* (45)*. Then for any $(\mathbf{z}, \mathbf{w}) \in \mathbb{R}^{\mathrm{n}}_{\geq 0} \times \mathbb{R}^{\mathrm{n}}_{\geq 0}$, we have:*

$$\tilde{\ell}(\mathbf{z}, \mathbf{z}) + \tilde{\ell}(\mathbf{w}, \mathbf{w}) = 2\tilde{\ell}(\mathbf{z}, \mathbf{w}) + (\mathbf{z} - \mathbf{w})^T \mathbf{Q}^- (\mathbf{z} - \mathbf{w}). \tag{70}$$

Proof We proceed by directly computing:

$$\begin{aligned}
\tilde{\ell}(\mathbf{z}, \mathbf{z}) + \tilde{\ell}(\mathbf{w}, \mathbf{w}) &= f(\mathbf{z}) + g(\operatorname{supp}(\mathbf{z})) + f(\mathbf{w}) + g(\operatorname{supp}(\mathbf{w})) \\
&= \mathbf{z}^T \mathbf{Q}^+ \mathbf{z} + \mathbf{z}^T \mathbf{Q}^- \mathbf{z} + \mathbf{z}^T \mathbf{p} + \mathbf{w}^T \mathbf{Q}^+ \mathbf{w} + \mathbf{w}^T \mathbf{Q}^- \mathbf{w} + \mathbf{w}^T \mathbf{p} \\
&\quad + g(\operatorname{supp}(\mathbf{z})) + g(\operatorname{supp}(\mathbf{w})).
\end{aligned}$$

Then, adding and subtracting the missing cross term, we have:

$$\begin{aligned}
\tilde{\ell}(\mathbf{z}, \mathbf{z}) + \tilde{\ell}(\mathbf{w}, \mathbf{w}) &= \mathbf{z}^T \mathbf{Q}^+ \mathbf{z} + \mathbf{w}^T \mathbf{Q}^+ \mathbf{w} + \mathbf{z}^T \mathbf{p} + \mathbf{w}^T \mathbf{p} + g(\operatorname{supp}(\mathbf{z})) + g(\operatorname{supp}(\mathbf{w})) \\
&\quad + \mathbf{z}^T \mathbf{Q}^- \mathbf{z} + \mathbf{w}^T \mathbf{Q}^- \mathbf{w} \\
&= 2\tilde{\ell}(\mathbf{z}, \mathbf{w}) + \mathbf{z}^T \mathbf{Q}^- \mathbf{z} - 2\mathbf{z}^T \mathbf{Q}^- \mathbf{w} + \mathbf{w}^T \mathbf{Q}^- \mathbf{w} \\
&= 2\tilde{\ell}(\mathbf{z}, \mathbf{w}) + (\mathbf{z} - \mathbf{w}) \mathbf{Q}^- (\mathbf{z} - \mathbf{w})
\end{aligned}$$

■

We also provide a proof that the condition on the minimizers of the lifted problem is not only sufficient, but necessary.

Lemma 26 *If $(\mathbf{z}^*, \mathbf{z}^*)$ and $(\mathbf{w}^*, \mathbf{w}^*)$ are minimizers of the lifted problem* (45)*, then:*

$$(\mathbf{z}^* - \mathbf{w}^*)^T \mathbf{Q}^- (\mathbf{z}^* - \mathbf{w}^*) \leq 0.$$

Proof Note that by the submodularity of $\tilde{\ell}$, if $(\mathbf{z}^*, \mathbf{z}^*)$ and $(\mathbf{w}^*, \mathbf{w}^*)$ are minimizers of the lifted problem (45), then so are their join $(\mathbf{z}^* \curlyvee \mathbf{w}^*, \mathbf{z}^* \curlywedge \mathbf{w}^*)$ and meet $(\mathbf{z}^* \curlywedge \mathbf{w}^*, \mathbf{z}^* \curlyvee \mathbf{w}^*)$. Then, working through the proof of Lemma 18 backwards proves the result. ■

References

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

Francis Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1-2):419–459, 2019.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

Francis R Bach. Shaping level sets with submodular functions. In *Advances in Neural Information Processing Systems*, pages 10–18, 2011.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

An Bian, Kfir Y. Levy, Andreas Krause, and Joachim M. Buhmann. Non-monotone continuous dr-submodular maximization: Structure and algorithms. *CoRR*, abs/1711.02515, 2017. URL `http://arxiv.org/abs/1711.02515`.

Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. *CoRR*, abs/1606.05615, 2016. URL `http://arxiv.org/abs/1606.05615`.

Jonathan Borwein and Adrian Lewis. *Convex Analysis and Nonlinear Optimization : Theory and Examples*. Number 2 in CMS Books in Mathematics. Springer-Verlag New York, 2006. ISBN 978-0-387-31256-9. doi: 10.1007/978-0-387-31256-9.

R. Bro and S. De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997. doi: https://doi.org/10.1002/(SICI)1099-128X(199709/10)11:5⟨393::AID-CEM483⟩3.0.CO;2-L.

E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. doi: 10.1109/TIT.2005.858979.

D. Chen, S.L. Sain, and K. Guo. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19:197–208, 2012.

A. Das and D. Kempe. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *The Journal of Machine Learning Research*, 19(1):74–107, 2018.

BA Davey and HA Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Marwa El Halabi and Stefanie Jegelka. Optimal approximation for unconstrained non-submodular minimization. *International Conference on Machine Learning (ICML)*, 2020.

Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B): 3539–3568, 2018.

Satoru Fujishige and Shigueo Isotani. A submodular function minimization algorithm based on the minimum-norm base. *Pacific Journal of Optimization*, 7(1):3–17, 2011.

Satoru Fujishige, Tamás Király, Kazuhisa Makino, Kenjiro Takazawa, and Shin ichi Tanigawa. Minimizing submodular functions on diamonds via generalized fractional matroid matchings. *Journal of Combinatorial Theory, Series B*, 157:294–345, 2022. ISSN 0095-8956. doi: https://doi.org/10.1016/j.jctb.2022.07.005. URL `https://www.sciencedirect.com/science/article/pii/S0095895622000715`.

S. Ito and R. Fujimaki. Large-scale price optimization via network flow. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/cb79f8fa58b91d3af6c9c991f63962d3-Paper.pdf`.

S. Ito and R. Fujimaki. Optimization beyond prediction: Prescriptive price optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1833–1841, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098188. URL `https://doi.org/10.1145/3097983.3098188`.

Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Fast semidifferential-based submodular function optimization: Extended version. In *International Conference on Machine Learning (ICML)*, 2013.

Sunyoung Kim and Masakazu Kojima. Exact solutions of some nonconvex quadratic optimization problems via sdp and socp relaxations. *Computational Optimization and Applications*, 26(2):143–154, 2003. doi: 10.1023/A:1025794313696. URL `https://doi.org/10.1023/A:1025794313696`.

Andreas Krause. Sfo: A toolbox for submodular function optimization. *Journal of Machine Learning Research (JMLR)*, 11(Mar):1141–1144, 2010.

Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *International Conference on Information Processing in Sensor Networks*, pages 2–10, 2006.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, USA, 2011. Association for Computational Linguistics. ISBN 9781932432879.

László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

K. Nagano, Y. Kawahara, and K. Aihara. Size-constrained submodular minimization through minimum norm base. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 977–984, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1): 265–294, 1978.

A. Qian, S. Zhu, J. Tang, R. Jin, B. Sun, and H. Li. Robust optimization over multiple domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4739–4746, 2019.

Holger Rauhut. Compressive sensing and structured random matrices. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, 9:1–92, 2010.

Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. *Applied Mathematics & Optimization*, 2019. doi: 10.1007/s00245-019-09567-0. URL `https://doi.org/10.1007/s00245-019-09567-0`.

Donald M Topkis. *Supermodularity and complementarity*. Princeton University Press, 1998.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research (JMLR)*, 7(Nov):2541–2563, 2006.