Approximation Guarantees of Local Search Algorithms via Localizability of Set Functions

Kaito Fujii [1]

Abstract

This paper proposes a new framework for providing approximation guarantees of local search algorithms. Local search is a basic algorithm design technique and is widely used for various combinatorial optimization problems. To analyze local search algorithms for set function maximization, we propose a new notion called *localizability* of set functions, which measures how effective local improvement is. Moreover, we provide approximation guarantees of standard local search algorithms under various combinatorial constraints in terms of localizability. The main application of our framework is sparse optimization, for which we show that restricted strong concavity and restricted smoothness of the objective function imply localizability, and further develop accelerated versions of local search algorithms. We conduct experiments in sparse regression and structure learning of graphical models to confirm the practical efficiency of the proposed local search algorithms.

1. Introduction

Local search is a widely used technique to design efficient algorithms for optimization problems. Roughly speaking, local search algorithms start with an initial solution and gradually increase the objective value by repeatedly moving the solution to a nearby point. While this approach leads to effective heuristics for many optimization problems in practice, it is not always easy to provide approximation guarantees on their performance.

In this paper, we propose a generic framework for providing approximation guarantees of local search algorithms for *set function optimization*. Set function optimization is a problem of finding an (approximately) optimal set from all feasible sets. Various machine learning tasks have been formulated as set function optimization problems, such as feature selection (Das & Kempe, 2011; Elenberg et al., 2018), summarization (Lin & Bilmes, 2011; Badanidiyuru et al., 2014), or active learning (Hoi et al., 2006; Golovin & Krause, 2011).

A promising approach to analyze local search algorithms for set function optimization is to utilize *submodularity*. Submodularity (Fujishige, 2005) is a property of set functions useful for designing efficient algorithms and has been extensively studied. Existing studies showed that for maximizing a submodular function subject to a certain constraint, local search procedures yield a constant-factor approximate solution to an optimal solution (Nemhauser et al., 1978; Fisher et al., 1978; Lee et al., 2010; Feldman et al., 2011). Local search algorithms have been applied to several machine learning tasks that have submodularity (Iyer et al., 2013; Balkanski et al., 2016), but there are many other practical set functions that deviate from submodularity.

To analyze local search algorithms for these problems, we propose a novel analysis framework that can be applied to local search algorithms for non-submodular functions. The key notion of our framework is a property of set functions, which we call *localizability*. Intuitively, localizability is a property that implies any local optimum is a good approximation to a global optimal solution. By utilizing this property, we show that for maximizing a set function with localizability under a certain constraint, simple local search algorithms achieve a good approximation.

The main application of our framework is *sparse optimization*. Sparse optimization is the problem of finding a sparse vector that optimizes a continuous objective function. It has various applications such as feature selection for sparse regression and structure learning of graphical models. An approach to sparse optimization is a reduction to a set function optimization problem, which is adopted by Jain et al. (2016) and Elenberg et al. (2017). We show that localizability of this set function is derived from restricted strong concavity and restricted smoothness of the original objective function, which implies approximation guarantees of local search algorithms. Furthermore, we devise accelerated

[1]National Institute of Informatics, Tokyo, Japan. Correspondence to: Kaito Fujii <fujiik@nii.ac.jp>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Table 1. Comparison of existing bounds on approximation ratios of local search algorithms, greedy algorithms, and modular approximation for sparse optimization with combinatorial constraints. The result of Elenberg et al. (2017) is indicated by †. The result of Chen et al. (2018) is indicated by ‡. M_s and $M_{s,t}$ are restricted smoothness constants and m_s is a restricted strong concavity constant (See Definition 4 for details). T is the number of iterations of local search algorithms and s is the maximum cardinality of feasible solutions.

Constraint	Local search	Greedy-based	Modular approx.
Cardinality	$\frac{m_{2s}^2}{M_{s,2}^2}\left(1 \quad \exp\left(\frac{M_{s,2}T}{sm_{2s}}\right)\right)$	$1 \quad \exp\left(\quad \frac{m_{2s}}{M_{s,1}}\right)$ †	$\frac{m_1 m_s}{M_1 M_s}$
Matroid	$\frac{m_{2s}^2}{M_{s,2}^2}\left(1 \quad \exp\left(\frac{M_{s,2}T}{sm_{2s}}\right)\right)$	$\frac{1}{(1+\frac{M_{s,1}}{m_s})^2}$ ‡	$\frac{m_1 m_s}{M_1 M_s}$
p-Matroid intersection or p-Exchange systems	$\frac{1}{p \quad 1+1/q}\frac{m_{2s}^2}{M_{s,2}^2}\left(1 \quad \exp\left(\frac{(p \quad 1+1/q)M_{s,2}T}{sm_{2s}}\right)\right)$	N/A	$\frac{1}{p \quad 1+1/q}\frac{m_1 m_s}{M_1 M_s} \quad \epsilon$

variants of our proposed local search algorithms by utilizing the structure of sparse optimization. An advantage of our approach over existing methods is its applicability to a broader class of combinatorial constraints.

Our contribution. In this paper, we propose a new property of set functions called *localizability* and provide a lower bound on the approximation ratio of local search algorithms for maximizing a set function with this property. Our contribution is summarized as follows.

- We define localizability of set functions and show that localizability of sparse optimization is derived from restricted strong concavity and restricted smoothness of the original objective function.
- Under the assumption of localizability, we provide lower bounds on the approximation ratio of a standard local search algorithm under a matroid constraint, p-matroid intersection constraint, or p-exchange system constraint.
- For sparse optimization, we propose two accelerated variants of local search algorithms, which we call *semi-oblivious* and *non-oblivious* local search algorithms.
- We conduct experiments on sparse regression and structure learning of graphical models to confirm the practical efficiency of our accelerated local search algorithms.

1.1. Related Work

Local search for submodular maximization. For monotone submodular maximization, several algorithms have been designed based on local search. Nemhauser et al. (1978) proposed a $1/2$-approximation local search procedure for a cardinality constraint, which they call an *interchange heuristic*. Fisher et al. (1978) generalized this result to a single matroid constraint. For a p-matroid intersection constraint, Lee et al. (2010) proposed a $(1/p \quad \epsilon)$-approximation local search algorithm. Feldman et al. (2011) proposed a novel class of constraints called p-exchange systems and devised a $(1/p \quad \epsilon)$-approximation local search algorithm. Filmus & Ward (2014) devised a $(1 \quad 1/\mathrm{e})$-approximation local search algorithm for a matroid constraint. Also for non-monotone submodular maximization, constant-factor approximation local search algorithms have been devised (Feige et al., 2011; Lee et al., 2009). While local search algorithms for submodular maximization have been studied extensively, there are only a few results on those for non-submodular maximization.

Approximation algorithms for non-submodular function maximization and sparse optimization. For non-submodular function maximization, many existing studies have adopted an approach based on *greedy algorithms*. Das & Kempe (2011) analyzed greedy algorithms for sparse linear regression by introducing the notion of submodularity ratio. Elenberg et al. (2018) extended their results to sparse optimization problems with restricted strong concavity and restricted smoothness. Chen et al. (2018) showed that the random residual greedy algorithm achieves $(\quad /(1+ \quad))^2$-approximation for a set function maximization with submodularity ratio under a single matroid constraint[1]. We compare our results with existing methods for sparse optimization in Table 1. Note that the results of Das & Kempe (2011) and Chen et al. (2018) hold for any monotone set function whose submodularity ratio is bounded, while we utilize a stronger property derived from restricted strong concavity and restricted smoothness. Bian et al. (2017) analyzed the greedy algorithm for maximizing a set function whose submodularity ratio and generalized curvature are both bounded. Sakaue (2019) considered sparse optimization with a constraint expressed by a monotone set function with bounded superadditivity ratio and restricted inverse curvature, but their framework cannot deal with matroid constraints and p-exchange system constraints. Fujii & Soma (2018) developed a similar analysis to ours, but

[1]They did not specify the subscripts of γ, but it is not larger than $\min_{i=1,\cdots,s}\gamma_{i-1,s-i}$, where s is the rank of the matroid.

their focus lies in a different problem called dictionary selection. The Frank-Wolfe algorithm (Frank & Wolfe, 1956; Jaggi, 2013) is a continuous optimization method that is often applied to sparse optimization. A variant called the *pairwise Frank-Wolfe algorithm* (Lacoste-Julien & Jaggi, 2015) incorporates the technique of moving weight between two atoms at each iteration, which is similar to our local search procedures, but their guarantees are incomparable to ours.

Modular approximation for sparse optimization. For sparse optimization with structured constraints, there exists a trivial benchmark called *modular approximation* (Cevher & Krause, 2011). Modular approximation maximizes a linear function that approximates the original set function by ignoring all interactions between elements. We provide a detailed description and analysis of modular approximation in Appendix D.

Sparse recovery. Many existing studies on sparse optimization focus on sparse recovery guarantees, which cannot be compared directly with our guarantees on approximation ratios. In the context of compressed sensing, several algorithms similar to our non-oblivious local search algorithms have been developed (Needell & Tropp, 2010; Bahmani et al., 2013). Kyrillidis & Cevher (2012) and Baldassarre et al. (2016) developed frameworks that can be applied to a matroid constraint. For structure learning of graphical models, Jalali et al. (2011) provided sparse recovery guarantees for the forward-backward greedy algorithm by assuming restricted strong concavity and restricted smoothness. Recently, algorithms with recovery guarantees under weaker assumptions have been developed (Bresler, 2015; Klivans & Meka, 2017; Wu et al., 2019).

1.2. Organization

The rest of this paper is organized as follows. Section 2 specify the problem settings that we tackle in this paper. Section 3 introduces the notion of localizability and shows localizability of sparse optimization. In Section 4, we propose local search algorithms for a matroid constraint, p-matroid intersection constraint, or p-exchange system constraint. In Section 5, we devise accelerated local search algorithms for sparse optimization. In Section 6, we describe applications of our problem settings: sparse regression and structure learning of graphical models. In Section 7, we empirically compare our proposed algorithms with existing methods. Due to space constraints, we defer all proofs to the appendix.

2. Problem Setting

In this section, we introduce the problem settings that we deal with in this paper.

Set function maximization. Let $N \coloneqq [n]$ be the ground set and define a non-negative set function $f\colon 2^N \to \mathbb{R}_{\geq 0}$. Throughout the paper, we assume f is monotone, i.e., $f(X) \leq f(Y)$ holds for any $X \subseteq Y \subseteq N$. We say f is submodular when $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$ for any $S \subseteq T \subseteq N$ and $v \in N \setminus T$. Let $\mathcal{I} \subseteq 2^N$ be a set family that represents all feasible solutions. We assume $(N, \mathcal{I})$ is an independence system, that is, $\emptyset \in \mathcal{I}$ and $X \in \mathcal{I}$ for any $X \subseteq Y$ such that $Y \in \mathcal{I}$. A set function maximization problem can be written as

$$\text{Maximize} \quad f(X) \qquad \text{subject to} \quad X \in \mathcal{I}. \tag{1}$$

In general, suppose we have access to a value oracle and independence oracle, which return the value of $f(X)$ and the Boolean value that represents whether $X \in \mathcal{I}$ or not for any input $X \in N$, respectively.

We consider three classes of independence systems: matroid constraints, p-matroid intersection, and p-exchange systems, which include structures that appear in applications. A standard setting of sparse optimization where $\mathcal{I} = \{X \subseteq N \mid |X| \leq s\}$ is a special case of matroid constraints.

Definition 1 (Matroids)**.** An independence system $(N, \mathcal{I})$ is called a *matroid* if for any $S, T \in \mathcal{I}$ with $|S| < |T|$, there exists $v \in T \setminus S$ such that $S \cup \{v\} \in \mathcal{I}$.

Definition 2 (p-Matroid intersection)**.** An independence system $(N, \mathcal{I})$ is a *p-matroid intersection* if there exist p matroids $(N, \mathcal{I}_1), \cdots, (N, \mathcal{I}_p)$ such that $\mathcal{I} = \bigcap_{i=1}^{p} \mathcal{I}_i$.

Definition 3 (p-Exchange systems (Feldman et al., 2011))**.** An independence system $(N, \mathcal{I})$ is a *p-exchange system* if for any $S, T \in \mathcal{I}$, there exists a map $\varphi\colon (T \setminus S) \to 2^{S \setminus T}$ such that (a) for any $v \in T \setminus S$, it holds that $|\varphi(v)| \leq p$, (b) each $v \in S \setminus T$ appears in $(\varphi(v))_{v \in T \setminus S}$ at most p times, and (c) for any $X \subseteq T \setminus S$, it holds that $(S \setminus \bigcup_{v \in X} \varphi(v)) \cup X \in \mathcal{I}$.

Sparse optimization. Sparse optimization is the problem of finding a sparse solution that maximizes a continuously differentiable function $u\colon \mathbb{R}^n \to \mathbb{R}$. Assume we have an access to a zeroth and first-order oracle that returns the value of $u(\mathbf{w})$ and gradient $\nabla u(\mathbf{w})$ given $\mathbf{w} \in \mathbb{R}^n$. To define the approximation ratio properly, we need to assume $u(\mathbf{0}) \neq 0$, but we can normalize any function $u'\colon \mathbb{R}^n \to \mathbb{R}$ by setting $u(\mathbf{w}) \coloneqq u'(\mathbf{w}) - u'(\mathbf{0})$. Let $N = [n]$ be the set of all variables and $\mathcal{I} \subseteq 2^N$ a family of feasible supports. We can write a sparse optimization problem with structured constraints as

$$\text{Maximize} \quad u(\mathbf{w}) \qquad \text{subject to} \quad \mathrm{supp}(\mathbf{w}) \in \mathcal{I}, \tag{2}$$

where $\mathrm{supp}(\mathbf{w})$ represents the set of non-zero elements of $\mathbf{w}$, that is, $\mathrm{supp}(\mathbf{w}) = \{i \in N \mid \mathbf{w}_i \neq 0\}$. We define $\|\mathbf{w}\|_0 = |\mathrm{supp}(\mathbf{w})|$.

We assume restricted strong concavity and restricted smoothness of the objective function u, which are defined as follows.

Definition 4 (Restricted strong concavity and restricted smoothness (Negahban et al., 2012; Jain et al., 2014))**.** Let Ω be a subset of $\mathbb{R}^d \times \mathbb{R}^d$ and $u\colon \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. We say that u is *restricted strongly concave* with parameter m_Ω and *restricted smooth* with parameter M_Ω on domain Ω if

$$\frac{m_\Omega}{2}\|\mathbf{y} \quad \mathbf{x}\|_2^2 \quad u(\mathbf{y}) \quad u(\mathbf{x}) - \langle \nabla u(\mathbf{x}), \mathbf{y} \quad \mathbf{x}\rangle \quad \frac{M_\Omega}{2}\|\mathbf{y} \quad \mathbf{x}\|_2^2$$

for all $(\mathbf{x}, \mathbf{y}) \in \Omega$.

Let $\Omega_s = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n \mid \|\mathbf{x}\|_0 \leq s, \|\mathbf{y}\|_0 \leq s, \|\mathbf{x} \quad \mathbf{y}\|_0 \leq s\}$ and $\Omega_{s,t} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n \mid \|\mathbf{x}\|_0 \leq s, \|\mathbf{y}\|_0 \leq s, \|\mathbf{x} \quad \mathbf{y}\|_0 \leq t\}$. Let m_s be restricted strong concavity parameter on Ω_s and $M_{s,t}$ the restricted smoothness parameter on $\Omega_{s,t}$ for any positive integer $s, t \in \mathbb{Z}_{>0}$. Due to the restricted strong concavity of u, $\operatorname{argmax}_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$ is uniquely determined. We denote this maximizer by $\mathbf{w}^{(X)}$.

By introducing a set function $f\colon 2^N \to \mathbb{R}_{\quad 0}$ defined as

$$f(X) = \max_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w}),$$

we can regard the sparse optimization problem as a set function optimization problem (1).

Notations. Vectors are denoted by bold lower-case letters (e.g. $\mathbf{x}$ and $\mathbf{y}$) and matrices are denoted by bold upper-case letters (e.g. $\mathbf{A}$ and $\mathbf{B}$). Sets are denoted by upper-case letters (e.g. X and Y). For $X \subseteq N$ and $a \in N$, we define $X + a \coloneqq X \cup \{a\}$ and $X \quad a \coloneqq X \setminus \{a\}$. We define the symmetric difference by $X \triangle Y \coloneqq (X \setminus Y) \cup (Y \setminus X)$.

3. Localizability of Set Functions

In this section, we define a property of set functions, which we call *localizability*. Since the general version of the definition of localizability is complicated, we first introduce a simplified definition as a warm-up, and then state the general definition.

Intuitively, localizability measures how much small modifications of a solution increase the objective value. Localizability is defined as a property that the sum of the increases yielded by small modifications is no less than the increase yielded by a large modification.

Definition 5 (Localizability (simplified version))**.** Let $f\colon 2^N \to \mathbb{R}_{\quad 0}$ be a non-negative monotone set function. For some $\alpha, \quad \in \mathbb{R}_{\quad 0}$, we say f is $(\alpha, \quad)$*-localizable with size* s if for arbitrary subsets $X, X^* \subseteq N$ of size s and bijection $\quad : X \setminus X^* \to X^* \setminus X$, we have

$$\sum_{x \in X \setminus X^*} \{f(X \quad x + \quad (x)) \quad f(X)\} \quad \alpha f(X^*) \quad f(X).$$

This property is sufficient to provide an approximation guarantee of local search algorithms for matroid constraints. However, we need a generalized version of localizability that considers exchanges of multiple elements to deal with more complicated constraints.

Definition 6 (Localizability)**.** Let $f\colon 2^N \to \mathbb{R}_{\quad 0}$ be a non-negative monotone set function. For some $\alpha, \quad_1, \quad_2 \in \mathbb{R}_{\quad 0}$, we say f is $(\alpha, \quad_1, \quad_2)$*-localizable with size* s *and exchange size* t if for arbitrary subsets $X, X^* \subseteq N$ of size at most s and any collection $\mathcal{P}$ of subsets of $X \triangle X^*$ such that $|P| \leq$ for each $P \in \mathcal{P}$, we have

$$\sum_{P \in \mathcal{P}} \{f(X \triangle P) \quad f(X)\} \quad \alpha k f(X^*) \quad (\quad_1 \ell + \quad_2 k) f(X),$$

where k and ℓ are positive integers such that each element in $X^* \setminus X$ appears at least k times in $\mathcal{P}$ and each element in $X \setminus X^*$ appears at most ℓ times in $\mathcal{P}$.

If we consider the case when $k = 1$ and $\ell = 1$, this definition coincides with the simplified version with $\quad = \quad_1 + \quad_2$. In existing studies on submodular maximization, Lee et al. (2010) and Feldman et al. (2011) utilized this property of linear functions and non-negative monotone submodular functions to prove the approximation bounds for local search algorithms.

Proposition 7 (Proved in the proof of Lemma 3.1 of Lee et al. (2010))**.** *Any linear function is* $(1, 1, 0)$*-localizable and any non-negative monotone submodular function is* $(1, 1, 1)$*-localizable with any size and any exchange size.*

In the following proposition, we show that the set function derived from sparse optimization satisfies localizability under the restricted strong concavity and restricted smoothness assumption.

Proposition 8. *Suppose* $u\colon 2^N \to \mathbb{R}$ *is a continuously differentiable function with* $u(\mathbf{0}) \quad 0$*. Let* $s, t \in \mathbb{Z}_{\quad 0}$ *be arbitrary integers. Assume* u *is restricted strong concave on* Ω_{2s} *and restricted smooth on* $\Omega_{s,t}$*. If* $f\colon 2^N \to \mathbb{R}$ *is a set function defined as* $f(X) = \max_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$*, then* f *is* $\left(\frac{m_{2s}}{M_{s,t}}, \frac{M_{s,t}}{m_{2s}}, 0\right)$*-localizable with size* s *and exchange size* t*.*

4. Local Search Algorithms

In this section, we describe our proposed local search algorithms for a matroid constraint, a p-matroid intersection constraint, and a p-exchange system constraint, and provide approximation ratio bounds in terms of localizability.

Algorithm 1 Local search algorithms for a matroid constraint

Let $X \leftarrow \emptyset$.
Add arbitrary elements to X until X is maximal in $\mathcal{I}$.
for $i = 1, \cdots, T$ **do**
  Find the pair of $x \in X$ and $x' \in N \setminus X$ such that $(x, x') \in \operatorname{argmax}\{f(X \setminus x + x') \mid X \setminus x + x' \in \mathcal{I}\}$
  if $f(X \setminus x + x') > f(X)$ **then**
    Update the solution $X \leftarrow X \setminus x + x'$.
  else
    return X.
  end if
end for
return X.

4.1. Algorithms for a Matroid Constraint

Here, we describe our proposed algorithm for a matroid constraint. The algorithm starts with an initial solution, which is any base of the given matroid. The main procedure of the algorithm is to repeatedly improve the solution by replacing an element in the solution with another element. At each iteration, the algorithm seeks a pair of an element $x \in X$ and another element $x' \in N \setminus X$ that maximizes $f(X \setminus x + x')$ while keeping the feasibility, which requires $\mathrm{O}(sn)$ oracle calls. The detailed description of the algorithms is given in Algorithm 1.

We can provide an approximation guarantee in terms of localizability of the objective function as follows.

Theorem 9. *Suppose $\mathcal{I}$ is the independence set family of a matroid and $s = \max\{|X| \mid X \in \mathcal{I}\}$. Assume the objective function f is non-negative, monotone, and $(\alpha, \varepsilon_1, \varepsilon_2)$-localizable with size s and exchange size 2. If X is the solution obtained by executing T iterations of Algorithm 1 and X^* is an optimal solution, then we have*

$$f(X) \geq \frac{\alpha}{\varepsilon_1 + \varepsilon_2}\left(1 - \exp\left(-\frac{(1 - \varepsilon_1 + \varepsilon_2)T}{s}\right)\right) f(X^*).$$

If X is the output returned by Algorithm 1 when it stops by finding no pair to improve the solution, then we have

$$f(X) \geq \frac{\alpha}{\varepsilon_1 + \varepsilon_2} f(X^*).$$

4.2. Algorithms for p-Matroid Intersection and p-Exchange System Constraints

In this section, we consider two more general constraints, p-matroid intersection and p-exchange system constraints with $p \geq 2$. The proposed algorithms for these two constraints can be described as almost the same procedure by using the different definitions of q-reachability as follows.

Definition 10 (q-Reachability for p-matroid intersection (Lee et al., 2010))**.** Let $\mathcal{I} \subseteq 2^N$ be a p-matroid intersection. A feasible solution $T \in \mathcal{I}$ is q-reachable from $S \in \mathcal{I}$ if $|T \setminus S| \leq 2q$ and $|S \setminus T| \leq 2pq$.

Algorithm 2 Local search algorithms for a p-matroid intersection or p-exchange system constraint ($p \geq 2$)

Let $X \leftarrow \emptyset$.
for $i = 1, \cdots, T$ **do**
  Find $X' \in \operatorname{argmax}_{X' \in \mathcal{F}_q(X)} f(X')$.
  if $f(X') > f(X) > 0$ **then**
    Update the solution $X \leftarrow X'$.
  else
    return X.
  end if
end for
return X.

Definition 11 (q-Reachability for p-exchange systems (Feldman et al., 2011))**.** Let $\mathcal{I} \subseteq 2^N$ be a p-exchange system. A feasible solution $T \in \mathcal{I}$ is q-reachable from $S \in \mathcal{I}$ if $|T \setminus S| \leq q$ and $|S \setminus T| \leq pq - q + 1$.

We denote by $\mathcal{F}_q(X)$ the set of all q-reachable sets from X that is determined by each definition of q-reachability for p-matroid intersection or p-exchange systems.

First, we must decide parameter $q \in \mathbb{Z}_{\geq 1}$ that determines the neighborhood to be searched at each iteration. When we select larger q, we search larger solution space for improvement at each iteration; thus, we can obtain a better bound on its approximation ratio, while the number of oracle calls $n^{\mathrm{O}(q)}$ becomes larger as well. The initial solution of the proposed algorithms is any feasible solution. Then the algorithms repeatedly replace the solution with a q-reachable solution that increases the objective value the most. The detailed description of this local search algorithm is given in Algorithm 2.

We can provide an approximation ratio bound under the assumption of localizability of the objective function as follows.

Theorem 12. *Suppose $\mathcal{I}$ is the independence set family of a p-matroid intersection or p-exchange system and $s = \max\{|X| \mid X \in \mathcal{I}\}$. Let $t = 2p(q+1)$ for the p-matroid intersection case and $t = pq + 1$ for the p-exchange system case. Assume the objective function f is non-negative, monotone, and $(\alpha, \varepsilon_1, \varepsilon_2)$-localizable with size s and exchange size t. If X is the output obtained by executing T iterations of Algorithm 2 with parameter q and X^* is an optimal solution, then the approximation ratio is lower-bounded by*

$$\frac{\alpha\left(1 - \exp\left(-\frac{(\beta_1(p - 1 + 1/q) + \beta_2)T}{s}\right)\right)}{\varepsilon_1(p - 1 + 1/q) + \varepsilon_2}.$$

If X is the output returned by Algorithm 2 when it stops by

finding no better q-reachable solution, then we have

$$f(X) \quad \frac{\alpha}{_1(p \quad 1+1/q) + \ _2} f(X^*).$$

5. Acceleration for Sparse Optimization

We consider two accelerated variants of the proposed local search algorithms in the case of sparse optimization. To distinguish the original one from the accelerated variants, we call Algorithm 1 and Algorithm 2 *oblivious local search algorithms*.

5.1. Acceleration for a Matroid Constraint

The oblivious version computes the value of $f(X \quad x+x')$ for $\mathrm{O}(sn)$ pairs of (x, x') at each iteration. We can reduce the computational cost by utilizing the structure of sparse optimization.

The first variant is the *semi-oblivious* local search algorithm. For each element $x' \in \not\in N \setminus X$ to be added, it computes the value of $f(X \quad x+x')$ only for $x \in \not\in X$ with the smallest $(\mathbf{w}^{(X)})^2_x$ among those satisfying $X \quad x+x' \in \not\in \mathcal{I}$. Thus, we can reduce the number of times we compute the value of $f(X \quad x+x')$ from $\mathrm{O}(sn)$ to $\mathrm{O}(n)$.

The second variant is the *non-oblivious* local search algorithm. It uses the value of

$$\frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x'} \quad \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_x$$

in place of the increase of the objective function $f(X \quad x+x') \quad f(X)$. We need to evaluate $\nabla u(\mathbf{w}^{(X)})$ and $\mathbf{w}^{(X)}$ at the beginning of each iteration, but it is not necessary to compute the value of $f(X \quad x+x')$.

The detailed description of these algorithms are given in Algorithm 3 in Appendix A.

Theorem 13. *Suppose $f(X) = \max_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$ and $\mathcal{I}$ is the independence set family of a matroid. If X is the solution obtained by executing T iterations of the semi-oblivious or non-oblivious local search algorithms and X^* is an optimal solution, then we have*

$$f(X) \quad \frac{m^2_{2s}}{M^2_{s,2}} \left(1 \quad \exp\left(\quad \frac{M_{s,2}T}{sm_{2s}}\right)\right) f(X^*),$$

where $s = \max\{|X| \colon X \in \not\in \mathcal{I}\}$. If X is the output returned when the algorithm stops by finding no pair to improve the solution, then we have

$$f(X) \quad \frac{m^2_{2s}}{M^2_{s,2}} f(X^*).$$

5.2. Acceleration for p-Matroid Intersection and p-Exchange System Constraints

Similarly to the case of matroid constraints, we can develop the semi-oblivious and non-oblivious local search algorithms for p-matroid intersection and p-exchange system constraints. The semi-oblivious variant only checks $X' \in \not\leftarrow \mathcal{F}_q(X)$ that minimizes $\mathbf{w}^{(X)}{}^2_{X \setminus X'}$ among $X'' \in \not\leftarrow \mathcal{F}_q(X)$ such that $X'' \setminus X = X' \setminus X$. The non-oblivious version selects the solution $X' \in \not\leftarrow \mathcal{F}_q(X)$ that maximizes

$$\frac{1}{2M_{s,t}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{X' \setminus X} \quad \frac{M_{s,t}}{2} \left(\mathbf{w}^{(X)}\right)^2_{X \setminus X'}.$$

The detailed description of these algorithms are given in Algorithm 4 in Appendix A. While the oblivious local search algorithm requires $O(n^q)$ times of the evaluation of f for finding the most suitable exchange at each iteration in general, the non-oblivious local search reduces it to a linear function maximization problem. In several cases such as a partition matroid constraint or a matching constraint, we can find the most suitable exchange in time polynomial in n and q by using standard techniques of combinatorial optimization. We can provide the same approximation guarantees for these accelerated variants as the oblivious variant.

Theorem 14. *Suppose $f(X) = \max_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$ and $\mathcal{I}$ is the independence set family of a p-matroid intersection or p-exchange system. Let $t = 2p(q+1)$ for the p-matroid intersection case and $t = pq+1$ for the p-exchange system case. If X is the output obtained by executing T iterations of the semi-oblivious or non-oblivious local search algorithms with parameter q and X^* is an optimal solution, then its approximation ratio is lower-bounded by*

$$\frac{1}{p \quad 1+1/q} \frac{m^2_{2s}}{M^2_{s,t}} \left(1 \quad \exp\left(\quad \frac{(p \quad 1+1/q)M_{s,t}T}{sm_{2s}}\right)\right),$$

where $s = \max\{|X| \colon X \in \not\in \mathcal{I}\}$. If X is the output returned when the algorithm stops by finding no better q-reachable solution, then we have

$$f(X) \quad \frac{1}{p \quad 1+1/q} \frac{m^2_{2s}}{M^2_{s,t}} f(X^*).$$

Remark 15. We also develop another version of our local search algorithms that increases the objective value at a predetermined rate, which is described in Appendix C.

Remark 16. The parameter $M_{s,t}$ used in the non-oblivious variant can be replaced with an upper bound on $M_{s,t}$, which leads to the approximation ratio bounds whose $M_{s,t}$ is also replaced with the upper bound.

6. Applications

In this section, we provide two applications of our framework: feature selection for sparse regression and structure

learning of graphical models.

6.1. Feature Selection for Sparse Regression

In sparse regression, given a design matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and a response vector $\mathbf{y}$, we aim to find a sparse vector $\mathbf{w} \in \mathbb{R}^n$ that optimizes some criterion. We can formulate this problem as a sparse optimization problem of maximizing $u(\mathbf{w})$ subject to $|\mathrm{supp}(\mathbf{w})| \leq s$, where $u \colon \mathbb{R}^n \to \mathbb{R}$ is the criterion determined by $\mathbf{A}$ and $\mathbf{y}$. Das & Kempe (2011) devised approximation algorithms in the case where u is the squared multiple correlation R^2, i.e., $u(\mathbf{w}) := 1 - \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 / \|\mathbf{y}\|_2^2$, and Elenberg et al. (2018) extended their results to general objectives with restricted strong concavity and restricted smoothness.

Here we consider sparse regression with *structured constraints*. In practical scenarios, we often have prior knowledge of relationships among features and can improve the quality of the estimation by incorporating it into structured constraints (Baraniuk et al., 2010; Huang et al., 2009). We formulate sparse regression with structured constraints as the problem of maximizing $u(\mathbf{w})$ subject to $\mathrm{supp}(\mathbf{w}) \in \mathcal{I}$, where $\mathcal{I}$ is the set family of feasible supports.

An advantage of our local search framework is its applicability to a broad class of structured constraints, including matroid constraints. For example, the following constraint is a special case of matroid constraints. Suppose the set of features are partitioned into several categories. Due to a balance among categories, it is often the case that we should select almost the equal number of features from each category. Such a constraint can be expressed by using a partition matroid. Partition matroid constraints were used for multi-level subsampling by Baldassarre et al. (2016) and detecting splice sites in precursor messenger RNAs by Chen et al. (2018). If there are multiple matroid constraints, we can formulate them as a p-matroid intersection constraint. To our knowledge, our proposed algorithms are the first to cope with multiple matroid constraints.

6.2. Structure Learning of Graphical Models

Undirected graphical models, or Markov random fields, express the conditional dependence relationships among random variables. We consider the problem of estimating the graph structure of an undirected graphical model given samples generated from this probability distribution. The goal of this problem is to restore the set of edges, that is, the set of all conditionally dependent pairs. To obtain a more interpretable graphical model, we often impose a sparsity constraint on the set of edges. This task can be formulated as a sparse optimization problem.

While most existing methods solve the neighborhood estimation problem separately for each vertex under a sparsity constraint (Jalali et al., 2011; Klivans & Meka, 2017), our framework provides an optimization method that handles the sparsity constraints for all vertices simultaneously. Suppose we aim to maximize some likelihood function (e.g., pseudo-log-likelihood (Besag, 1975)) under the sparsity constraint on each vertex, i.e., the degree of each vertex is at most b, where $b \in \mathbb{Z}_{\geq 0}$ is the maximum degree. This degree constraint is called a b-matching constraint, which is a special case of 2-exchange system. Hence, we can apply our local search algorithms to this problem.

7. Experiments

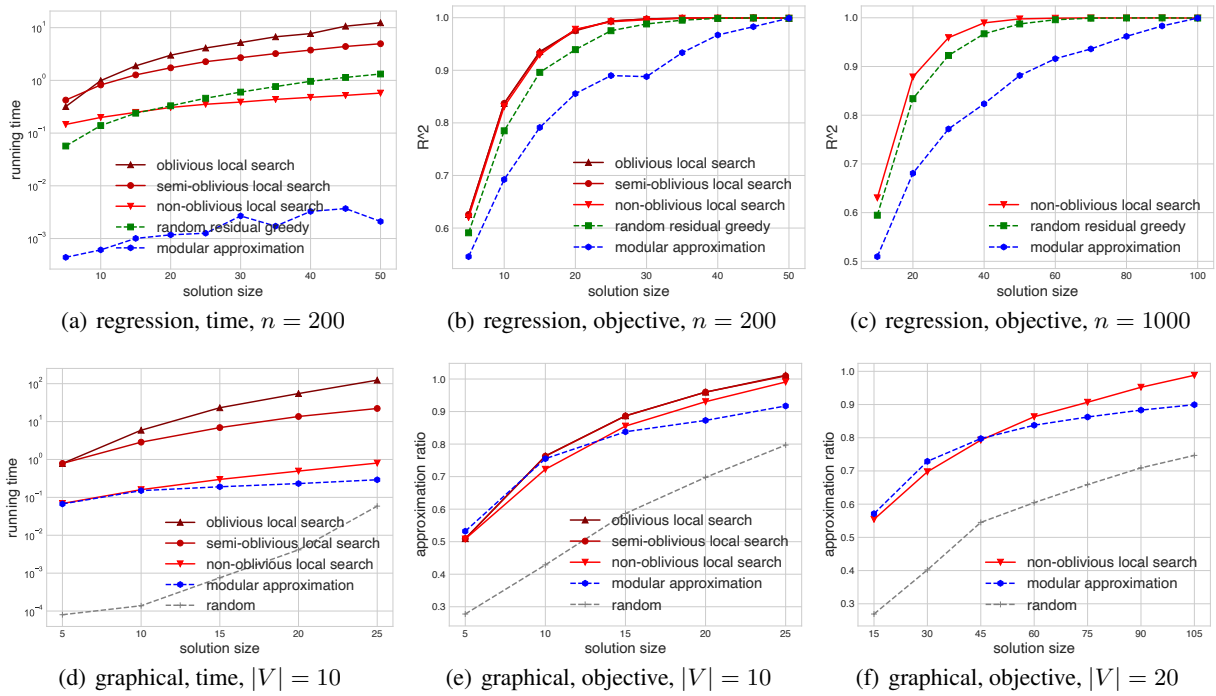
In this section, we conduct experiments on two applications: sparse regression and structure learning of graphical models. All the algorithms are implemented in Python 3.6. We conduct the experiments in a machine with Intel Xeon E3-1225 V2 (3.20 GHz and 4 cores) and 16 GB RAM.

7.1. Experiments on Sparse Regression

Datasets. We generate synthetic datasets with a partition matroid constraint. First, we determine the design matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ by generating each of its entries according to the uniform distribution on $[0, 1]$. Then we normalize each of columns to ensure that the mean will be 0 and the standard deviation will be 1. Suppose the set of all features are partitioned into n_c equal-size categories. We randomly select a sparse subset S^* by selecting n_p parameters from each category. The response vector is determined by $\mathbf{y} = \mathbf{A}_{S^*}\mathbf{w} + \epsilon$, where $\mathbf{w}$ is a random vector generated from the standard normal distribution $\mathcal{N}(0, 1)$ and ϵ is a noise vector whose each element is generated from $\mathcal{N}(0, 0.2)$. We consider two settings with different parameters. We set $(n, d, n_c, n_p) = (200, 50, 5, 5)$ in one setting and $(n, d, n_c, n_p) = (1000, 100, 10, 10)$ in the other setting. We use R^2 as the objective function to be maximized. For each parameter, we conduct 10 trials and plot the average.

Methods. We implement the oblivious, semi-oblivious, and non-oblivious local search algorithms. As benchmarks, we implement the random residual greedy algorithm (Chen et al., 2018) and modular approximation. We apply these methods to a partition matroid constraint with capacity n'_p for each $n'_p \in \{1, \cdots, 10\}$.

Results. First, we compare the proposed methods in the case of $n = 200$ (Figure 1(a) and Figure 1(b)). We can observe that the non-oblivious variant is approximately 10 times faster than the oblivious variant, while achieving an objective value comparable to that of the oblivious variant. In comparison to the random residual greedy algorithm, the non-oblivious variant achieves a higher objective value in a similar running time. Modular approximation is consid-



(a) regression, time, $n = 200$ (b) regression, objective, $n = 200$ (c) regression, objective, $n = 1000$

(d) graphical, time, $|V| = 10$ (e) graphical, objective, $|V| = 10$ (f) graphical, objective, $|V| = 20$

Figure 1. The experimental results on sparse linear regression under a partition matroid constraint (1(a), 1(b), and 1(c)) and structure learning of graphical models under a b-matching constraint (1(d), 1(e), and 1(f)). 1(a) shows the running time in the case where $n = 100$. 1(b) and 1(c) show the objective value (R^2) in the case where $n = 200$ and $n = 1000$, respectively. 1(d) shows the running time in the case where $|V| = 10$. 1(e) and 1(f) show the ratio between the objective achieved by the algorithms and the optimal objective value in the case where $|V| = 10$ and $|V| = 20$, respectively.

erably faster than the other methods, but the quality of its solution is poor. Next, we conduct experiments on larger datasets with $n = 1000$ (Figure 1(c)). The oblivious and semi-oblivious local search algorithms cannot be applied to this setting due to their slow running time. Moreover, in this setting, we can observe that the non-oblivious variant outperforms the benchmarks.

7.2. Experiments on Structure Learning of Graphical Models

Datasets. We consider Ising models $G = (V, E)$ with parameter $(w_{uv})_{u,v \in V}$. First we generate the true graphical model randomly from the configuration model with degree d for all vertices. For each edge $(u, v) \in E$, we set the parameter $w_{uv} = +0.5$ or $w_{uv} = -0.5$ uniformly at random. We synthetically generate 100 samples by Gibbs sampling from this Ising model. We consider two settings with different parameters. We set $(|V|, d) = (10, 5)$ in one setting and $(|V|, d) = (20, 7)$ in the other setting. We consider the problem of maximizing the pseudo-log-likelihood. For each setting, we conduct 10 trials and plot the average.

Methods. We implement the oblivious, semi-oblivious, and non-oblivious local search algorithms with parameter $q = 1$. Since calculating $M_{s,3}$ requires much computational cost, we use an upper bound $4 \sum_{i=1}^{N} \|\mathbf{x}^i\|_2^3$ instead of $M_{s,3}$ in the non-oblivious variant. As a benchmark, we implement modular approximation, in which to maximize a linear function over a b-matching constraint, we use the reduction from a max-weight b-matching problem to a max-weight matching problem (Schrijver, 2003, Theorem 32.4) and the max-weight matching problem solver in NetwerkX library. We also implement random selection, which randomly samples a subgraph whose degree is d at all vertices. In all methods, we use the L-BFGS-G solver in scipy.optimize library for evaluating the value of f. We apply these methods to pseudo-log-likelihood maximization under a b-matching constraint for each $b \in \{1, \cdots, d\}$.

Results. First, we compare the proposed methods in the case of $|V| = 10$ (Figure 1(d) and Figure 1(e)). We can observe that our acceleration techniques work well in practice. The running time of the non-oblivious variant is competitive with that of modular approximation and its solution quality is higher than that of modular approximation for larger solution size. Next, we conduct experiments on larger graphs with $|V| = 20$ (Figure 1(f)). Since the oblivious and semi-oblivious local search algorithms are too slow to be

applied to this setting, we omit them. Also, in this setting, we can observe that the non-oblivious variant outperforms the benchmarks particularly in cases of larger solution size.

Acknowledgements

The author would like to thank Andreas Krause for providing insightful comments in the early stages of this study. The author is thankful to Takeru Matsuda and Kazuki Matoya for inspiring discussions. This study was supported by JSPS KAKENHI Grant Number JP 18J12405.

References

Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., and Krause, A. Streaming submodular maximization: massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 671–680, 2014.

Bahmani, S., Raj, B., and Boufounos, P. T. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(1):807–841, 2013.

Baldassarre, L., Li, Y., Scarlett, J., Gozcu, B., Bogunovic, I., and Cevher, V. Learning-based compressive subsampling. *Journal of Selelected Topics in Signal Processing*, 10(4): 809–822, 2016.

Balkanski, E., Mirzasoleiman, B., Krause, A., and Singer, Y. Learning sparse combinatorial representations via two-stage submodular maximization. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pp. 2207–2216, 2016.

Baraniuk, R. G., Cevher, V., Duarte, M. F., and Hegde, C. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D*, 24(3):179–195, 1975.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 498–507, 2017.

Bresler, G. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC)*, pp. 771–782, 2015.

Cevher, V. and Krause, A. Greedy dictionary selection for sparse representation. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):979–988, 2011.

Chen, L., Feldman, M., and Karbasi, A. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 803–812, 2018.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 1057–1064, 2011.

Elenberg, E., Dimakis, A. G., Feldman, M., and Karbasi, A. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems (NIPS) 30*, pp. 4047–4057. 2017.

Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 46(6B):3539–3568, 2018.

Feige, U., Mirrokni, V. S., and Vondrák, J. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Feldman, M. *Maximization Problems with Submodular Objective Functions*. PhD thesis, Computer Science Department, Technion - Israel Institute of Technology, 2013.

Feldman, M., Naor, J., Schwartz, R., and Ward, J. Improved approximations for k-exchange systems (extended abstract). In *Proceedings of the 19th Annual European Symposium on Algorithms (ESA)*, pp. 784–798, 2011.

Filmus, Y. and Ward, J. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014.

Fisher, M. L., Nemhauser, G. L., and Wolsey, L. A. *An analysis of approximations for maximizing submodular set functions—II*, pp. 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Fujii, K. and Soma, T. Fast greedy algorithms for dictionary selection with generalized sparsity constraints. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pp. 4749–4758, 2018.

Fujishige, S. *Submodular Functions and Optimization*. Elsevier, 2nd edition, 2005.

Golovin, D. and Krause, A. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference of Machine Learning (ICML)*, pp. 417–424, 2006.

Huang, J., Zhang, T., and Metaxas, D. Learning with structured sparsity. *Journal of Machine Learning Research*, 12:3371–3412, 2009.

Iyer, R. K., Jegelka, S., and Bilmes, J. A. Fast semidifferential-based submodular function optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 855–863, 2013.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 427–435, 2013.

Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems (NIPS) 27*, pp. 685–693, 2014.

Jain, P., Rao, N., and Dhillon, I. S. Structured sparse regression via greedy hard thresholding. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1516–1524, 2016.

Jalali, A., Johnson, C. C., and Ravikumar, P. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems (NIPS) 24*, pp. 1935–1943, 2011.

Klivans, A. R. and Meka, R. Learning graphical models using multiplicative weights. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 343–354, 2017.

Kyrillidis, A. and Cevher, V. Combinatorial selection and least absolute shrinkage via the clash algorithm. In *Proceedings of the 2012 IEEE International Symposium on Information Theory (ISIT)*, pp. 2216–2220, 2012.

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems (NIPS) 28*, pp. 496–504, 2015.

Lee, J., Mirrokni, V. S., Nagarajan, V., and Sviridenko, M. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 323–332, 2009.

Lee, J., Sviridenko, M., and Vondrák, J. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4): 795–806, 2010.

Lin, H. and Bilmes, J. A. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pp. 510–520, 2011.

Needell, D. and Tropp, J. A. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.

Negahban, S. N., Ravikumar, P., Wainwright, M. J., and Yu, B. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1): 265–294, 1978.

Sakaue, S. Greedy and IHT algorithms for non-convex optimization with monotone costs of non-zeros. In *Proceedings of The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 206–215, 2019.

Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

Wu, S., Sanghavi, S., and Dimakis, A. G. Sparse logistic regression learns all discrete pairwise graphical models. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pp. 8069–8079, 2019.