Greed is Still Good: Maximizing Monotone Submodular+Supermodular (BP) Functions

Wenruo Bai [1] **Jeffrey A. Bilmes** [1 2]

Abstract

We analyze the performance of the greedy algorithm, and also a discrete semi-gradient based algorithm, for maximizing the sum of a suBmodular and suPermodular (BP) function (both of which are non-negative monotone non-decreasing) under two types of constraints, either a cardinality constraint or $p \geq 1$ matroid independence constraints. These problems occur naturally in several real-world applications in data science, machine learning, and artificial intelligence. The problems are ordinarily inapproximable to any factor. Using the curvature κ_f of the submodular term, and introducing κ^g for the supermodular term (a natural dual curvature for supermodular functions), however, both of which are computable in linear time, we show that BP maximization can be efficiently approximated by both the greedy and the semi-gradient based algorithm. The algorithms yield multiplicative guarantees of $\frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right]$ and $\frac{1-\kappa^g}{(1-\kappa^g)\kappa_f+p}$ for the two types of constraints respectively. For pure monotone supermodular constrained maximization, these yield $1-\kappa^g$ and $(1-\kappa^g)/p$ for the two types of constraints respectively. We also analyze the hardness of BP maximization and show that our guarantees match hardness by a constant factor and by $O(\ln(p))$ respectively. Computational experiments are also provided supporting our analysis.

1. Introduction

The *Greedy algorithm* (Bednorz, 2008; Cormen, 2009) is a technique in combinatorial optimization that makes a locally optimal choice at each stage in the hope of finding a good global solution. It is one of the simplest, most widely applied, and most successful algorithms in practice (Kempner et al., 2008; Zhang et al., 2000; Karp & Kung, 2000; Ruiz & Stützle, 2007; Wolsey, 1982). Due to its simplicity, and low time and memory complexities, it is used empirically even when no guarantees are known to exist although, being inherently myopic, the greedy algorithm's final solution can be arbitrarily far from the optimum solution (Bang-Jensen et al., 2004).

On the other hand, there are results going back many years showing where the greedy algorithm is, or almost is, optimal, including Huffman coding (Huffman, 1952), linear programming (Dunstan & Welsh, 1973; Dietrich & Hoffman, 2003), minimum spanning trees (Kruskal, 1956; Prim, 1957), partially ordered sets (Faigle, 1979; Dietrich & Hoffman, 2003), matroids (Edmonds, 1971; Dress & Wenzel, 1990), greedoids (Korte et al., 2012), and so on, perhaps culminating in the association between the greedy algorithm and submodular functions (Edmonds, 1970; Nemhauser et al., 1978a; Conforti & Cornuejols, 1984; Fujishige, 2005).

Submodular functions have recently shown utility for a number of machine learning and data science applications such as information gathering (Krause et al., 2006), document summarization (Lin et al., 2009; Lin & Bilmes, 2011a), image segmentation (Kohli et al., 2009; Jegelka & Bilmes, 2011), and string alignment (Lin & Bilmes, 2011b), since such functions are natural for modeling concepts such as diversity, information, and dispersion. Defined over an underlying ground set V, a set function $f : 2^V \to \mathbb{R}$ is said to be submodular when for all subsets $X, Y \subseteq V$, $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$. Defining $f(\{v\}|X) = f(\{v\} \cup X) - f(X)$ as the gain of adding the item v in the context of $X \subset V$, an equivalent characterization of submodularity is via *diminishing returns*: $f(\{v\}|X) \geq f(\{v\}|Y)$, for any $X \subseteq Y \subset V$ and $v \in V \setminus Y$. A set function f is monotonically non-decreasing if $f(\{v\}|S) \geq 0$ for all $v \in V \setminus S$ and it is normalized if $f(\emptyset) = 0$. In addition to being useful utility models, submodular functions also have amiable optimization properties — many submodular optimization problems (both maximization (Wolsey, 1982) and minimization (Cunningham, 1985)) admit polynomial time approximation or

*Equal contribution [1]Department of Electrical Engineering, University of Washington, Seattle, USA [2]Department of Computer Science and Engineering, University of Washington, Seattle, USA. Correspondence to: Wenruo Bai <wrbai@uw.edu>, Jeffrey A. Bilmes <bilmes@uw.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

exact algorithms. Most relevant presently, the greedy algorithm has a good constant-factor approximation guarantee, e.g., the classic $1 - 1/e$ and $1/(p + 1)$ guarantees for submodular maximization under a cardinality constraint or p matroid constraints (Nemhauser et al., 1978b; Fisher et al., 1978).

Certain subset selection problems in data science are not purely submodular, however. For example, when choosing a subset of training data in a machine learning system (Wei et al., 2015), there might be not only redundancies but also complementarities amongst certain subsets of elements, where the full collective utility of these elements is seen only when utilized together. Submodular functions can only diminish, rather than enhance, the utility of a data item in the presence of other data items. Supermodular set functions can model such phenomena, and are widely utilized in economics and social sciences, where the notion of complementary (Topkis, 2011) is naturally needed, but are studied and utilized less frequently in machine learning. A set function $g(X)$ is said to be supermodular if $-g(X)$ is submodular.

In this paper, we advance the state of the art in understanding when the greedy (and the semigradient) algorithm offers a guarantee, in particular for approximating the constrained maximization of an objective that may be decomposed into the sum of a submodular and a supermodular function (applications are given in Section 1.1). That is, we consider the following problem

$$\text{Problem 1.} \quad \max_{X \in \mathcal{C}} h(X) := f(X) + g(X), \qquad (1)$$

where $\mathcal{C} \subseteq 2^V$ is a family of feasible sets, f and g are normalized ($f(\emptyset) = 0$), monotonic non-decreasing ($f(\{s\} | S) \geq 0$ for any $s \in V$ and $S \subseteq V$) submodular and supermodular functions respectively[1] and hence are non-negative. We call this problem *suBmodular-suPermodular* (BP) maximization, and $f + g$ a ***BP function***, and we say h admits a BP decomposition if $\exists f, g$ such that $h = f + g$ where f and g are defined as above. In the paper, the set $\mathcal{C}$ may correspond either to a cardinality constraint (i.e., $\mathcal{C} = \{A \subseteq V \mid |A| \leq k\}$ for some $k \geq 0$), or alternatively, a more general case where $\mathcal{C}$ is defined as the intersection of p matroids. Hence, we may have $\mathcal{C} = \{X \subseteq V \mid X \in \mathcal{I}_1 \cap \mathcal{I}_2 \cap \cdots \cap \mathcal{I}_p\}$, where $\mathcal{I}_i$ is the set of independent sets for the ith matroid $\mathcal{M}_i = (V, \mathcal{I}_i)$. A *matroid* generalizes the concept of independence in vector spaces, and is a pair $(V, \mathcal{I})$ where V is the ground set and $\mathcal{I}$ is a family of subsets of V that are *independent* with the following three properties: (1) $\emptyset \in \mathcal{I}$; (2) $Y \in \mathcal{I}$ implies $X \in \mathcal{I}$ for all $X \subseteq Y \subseteq V$; and (3) if $X, Y \in \mathcal{I}$ and $|X| > |Y|$, then there exists $v \in X \setminus Y$ such that $Y \cup \{v\} \in \mathcal{I}$. Matroids are often used as combinatorial constraints, where a feasible set of an optimization problem must be independent in all p matroids.

The performance of the greedy algorithm for some special cases of BP maximization has been studied before. For example, when $g(X)$ is modular, the problem reduces to submodular maximization where, if f and g are also monotone, the greedy algorithm is guaranteed to obtain an $1 - 1/e$ approximate solution under a cardinality constraint (Nemhauser et al., 1978b) and $1/_{p+1}$ for p matroids (Fisher et al., 1978; Conforti & Cornuejols, 1984). The greedy algorithm often does much better than this in practice. Correspondingly, the bounds can be significantly improved if we also make further assumptions on the submodular function. One such assumption is the (total) *curvature*, defined as $\kappa_f = 1 - \min_{v \in V} \frac{f(v|V \setminus \{v\})}{f(v)}$ — the greedy algorithm has a $\frac{1}{\kappa_f}(1 - e^{-\kappa_f})$ and a $\frac{1}{\kappa_f + p}$ guarantee (Conforti & Cornuejols, 1984) for a cardinality and for p matroid constraints, respectively. Curvature is also attractive since it is linear time computable with only oracle function access. Liu et al. (2017) shows that κ_f can be replaced by a similar quantity, i.e., $b = 1 - \min_{v \in A \in \mathcal{I}} \frac{f(\{v\}|A \setminus \{v\})}{f(\{v\})}$ for a single matroid $\mathcal{M} = (V, \mathcal{I})$, a quantity defined only on the independent sets of the matroid, thereby improving the bounds further. In the present paper, however, we utilize the traditional definition of curvature. The current best guarantee is $1 - \kappa_f / e$ for a cardinality constraint using modifications of the continuous greedy algorithm (Sviridenko et al., 2015) and $\frac{1}{\epsilon + p}$ for multiple matroid constraints based on a local search algorithm (Lee et al., 2010). In another relevant result, Sarpatwar et al. (2017) gives a bound of $^{(1 - e^{-(p+1)})}/_{(p+1)}$ for submodular maximization with a single knapsack and the intersection of p matroid constraints.

When $g(X)$ is not modular, the problem is much harder and is NP-hard to approximate to any factor (Lemma 3.1). In our paper, we show that bounds are obtainable if we make analogous further assumptions on the supermodular function g. That is, we introduce a natural curvature notion to monotone non-decreasing nonnegative supermodular functions, defining the **supermodular curvature** as $\kappa^g = \kappa_{g(V) - g(V \setminus X)} = 1 - \min_{v \in V} \frac{g(v)}{g(v|V \setminus \{v\})}$. We note that κ^g is distinct from the steepness (Il'ev, 2001; Sviridenko et al., 2015) of a nonincreasing supermodular function (see Section 3.1). The function $g(V) - g(V \setminus X)$ is a normalized monotonic non-decreasing submodular function, known as the submodular function dual to the supermodular function g (Fujishige, 2005). *Supermodular curvature* is a natural dual to *submodular curvature* and, like submodular curvature, is computationally feasible to compute, requiring only linear time in the oracle model, unlike other measures of non-submodularity (Section 1.2). Hence, given a BP decomposition of $h = f + g$, it is possible, as we show below, to

[1]Throughout, f & g are assumed monotonic non-decreasing submodular/supermodular functions respectively.

	bound	hardness
cardinality	$\frac{1}{\kappa_f}\left[1-e^{-(1-\kappa^g)\kappa_f}\right]$	$1-\kappa^g+\epsilon$
p matroid	$\frac{1-\kappa^g}{(1-\kappa^g)\kappa_f+p}$	$(1-\kappa^g)O(\frac{\ln p}{p})$

Table 1. Lower bounds for GREEDMAX (Alg. 1)/SEMIGRAD (Alg. 2) and BP maximization hardness.

derive practical and useful quality assurances based on the curvature of each component of the decomposition.

We examine two algorithms, GREEDMAX (Alg. 1) and SEMIGRAD (Alg. 2) and show that, despite the two algorithms being different, both of them have a worst case guarantee of $\frac{1}{\kappa_f}\left[1-e^{-(1-\kappa^g)\kappa_f}\right]$ for a cardinality constraint (Theorem 3.6) and $\frac{1-\kappa^g}{(1-\kappa^g)\kappa_f+p}$ for p matroid constraints (Theorem 3.7). If $\kappa^g=0$ (i.e., g is modular), the bounds reduce to $\frac{1}{\kappa_f}(1-e^{-\kappa_f})$ and $\frac{1}{\kappa_f+p}$, which recover the aforementioned bounds. If $\kappa^g=1$ (i.e., g is fully curved) the bounds are 0 since, in general, the problem is NP-hard to approximate (Lemma 3.1). For pure monotone supermodular function maximization, the bounds yield $1-\kappa^g$ and $(1-\kappa^g)/p$ respectively. We also show that no polynomial algorithm can do better than $1-\kappa^g+\epsilon$ or $(1-\kappa^g)O(\frac{\ln p}{p})$ for cardinality or multiple matroid constraints respectively unless P=NP. Therefore, no polynomial algorithm can beat GREEDMAX by a factor of $\frac{1+\epsilon}{1-e^{-1}}$ or $O(\ln(p))$ for the two constraints unless P=NP.

1.1. Applications

Problem 1 naturally applies to a number of machine learning and data science applications.

Summarization with Complementarity Submodular functions are an expressive set of models for summarization tasks where they capture how data elements are mutually redundant. In some cases, however, certain subsets might be usefully chosen together, i.e., when their elements have a complementary relationship. For example, when choosing a subset of training data samples for supervised machine learning system (Wei et al., 2015), nearby points on opposite sides of a decision boundary would be more useful to characterize this boundary if chosen together. Also, for the problem of document summarization (Lin & Bilmes, 2011a; 2012), where a subset of sentences is chosen to represent a document, there are some cases where a single sentence makes sense only in the context of other sentences, an instance of complementarity. In such cases, it is reasonable to allow these relationships to be expressed via a monotone supermodular function. One such complementarity family takes g to be a weighted sum of monotone convex functions composed with non-negative modular functions, as in $g(A)=\sum_i w_i\psi_i(m_i(A))$. A still more expressive family includes the "deep supermodular functions" (Bilmes & Bai, 2017) which consist of multiple nested layers of such transformations. A natural formulation of the summarization-with-complementary problem is to maximize an objective that is the weighted sum of a monotone submodular utility function and one of the above complementarity functions. Hence, such a formulation is an instance of Problem 1. In either case, the supermodular curvature is easy to compute, and for many instances is less than unity leading to a quality assurance based on the results of this paper.

Generalized Bipartite Matching Submodularity has been used to generalize bipartite matching. For example, a generalized bipartite matching (Lin & Bilmes, 2011b) procedure starts with a non-negative weighted bipartite graph (V,U,E), where V is a set of left vertices, U is a set of right vertices, $E\subseteq V\times U$ is a set of edges, and $h:2^E\to\mathbb{R}_+$ is a score function on the edges. Note that a matching constraint is an intersection of two partition matroid constraints, so a matching can be generalized to the intersection of multiple matroid constraints. Word alignment between two sentences of different languages (Melamed, 2000) can be viewed as a matching problem, where each word pair is associated with a score reflecting the desirability of aligning that pair, and an alignment is formed as the highest scored matching under some constraints. Lin & Bilmes (2011b) uses a submodular objective functions that can represent complex interactions among alignment decisions. Also in (Bai et al., 2016), similar bipartite matching generalizations are used for the task of peptide identification in tandem mass spectrometry. By utilizing a BP function in Problem 1, our approach can extend this to allow also for complementarity to be represented amongst sets of matched vertices.

1.2. Approach, and Related Studies

An arbitrary set function can always be expressed as a difference of submodular (DS) functions (Narasimhan & Bilmes, 2005; Iyer & Bilmes, 2012a;c). Although finding such a decomposition itself can be hard (Iyer & Bilmes, 2012a), the decomposition allows for additional optimization strategies based on discrete semi-gradients (Equation (2)) that do not offer guarantees, even in the unconstrained case (Iyer & Bilmes, 2012a). Our problem is a special case of constrained DS optimization since a negative submodular function is supermodular. Our problem also asks for a BP decomposition of h which is not always possible even for monotone functions (Lemma 3.2). Constrainedly optimizing an arbitrary monotonic non-decreasing set function is impossible in polynomial time and not even approximable to any positive factor (Lemma 3.1). In general, there are two ways to approach such a problem: one is to offer polynomial time heuristics without any theoretical guarantee (and hence possibly performing arbitrarily poorly in worst case); another is to analyze (using possibly exponential time itself, e.g., see below starting with the submodularity ratio) the set function in order to provide theoretical guarantees. In our framework, as we will see, the BP decomposition not only allows for additional optimization strategies as does a DS decomposition, but also, given additional information about the curvature of the two components (computable easily in linear time), allows us to show how the set function can be approximately maximized in polynomial time with guarantees. With a curvature analysis, not only the greedy algorithm but also a semi-gradient optimization strategy (Alg. 2) attains a guarantee even in the constrained setting. We also argued, in Section 1.1, that BP functions, even considering their loss of expressivity relative to DS functions, are still quite natural in applications.

Algorithm 1: GREEDMAX for BP maximization

1: **Input:** f, g and constraint set $\mathcal{C}$.
2: **Output:** An approximation solution $\hat{X}$.
3: **Initialize:** $X_0 \leftarrow \emptyset$, $i \leftarrow 0$ and $R \leftarrow V$
4: **while** $\exists v \in R$ s.t. $X_i \cup v \in \mathcal{C}$ **do**
5: $\quad v \in \operatorname{argmax}_{v \in R, X_i \cup v \in \mathcal{C}} f(v|X_i) + g(v|X_i)$.
6: $\quad X_{i+1} \leftarrow X_i \cup v$, $R \leftarrow R \setminus v$.
7: $\quad i \leftarrow i + 1$.
8: **end while**
9: Return $\hat{X} \leftarrow X_i$.

Submodularity ratio and curvature Bian et al. (2017) offered a bound based on both the submodularity ratio and a newly introduced form of generalized curvature. The submodularity ratio (Das & Kempe, 2011) of a non-negative set function h is defined as the largest scalar γ s.t. $\sum_{\omega \in \Omega \setminus S} h(\omega|S) \geq \gamma h(\Omega|S), \forall \Omega, S \subseteq V$ and is equal to one if and only if h is submodular. It is often defined as $\gamma_{U,k}(h) = \min_{L \subseteq U, S:|S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} h(x|L)}{h(S|L)}$ for $U \subseteq V$ and $1 \leq k \leq |V|$, and then $\gamma = \gamma_{V,|V|}(h)$. The generalized curvature (Bian et al., 2017) of a non-negative set function h is defined as the smallest scalar α s.t. $h(i|S \setminus \{i\} \cup \Omega) \geq (1-\alpha) h(i|S \setminus \{i\}), \forall \Omega, S \subseteq V, i \in S \setminus \Omega$. (Bian et al., 2017) offers a lower bound of $\frac{1}{\alpha}(1 - e^{-\alpha\gamma})$ for the greedy algorithm. Computing this bound is not computationally feasible in general because both the submodularity ratio and the generalized curvature are information theoretically hard to compute under the oracle model, as we show in Section K.2. This is unlike curvatures κ_f, κ^g which are both computable in linear time given only oracle access to both f and g. We make further comparisons between the pair κ_f, κ^g with the submodularity ratio in Section K.

Approximately submodular functions A function h is said to be ϵ-approximately submodular if there exists a submodular function f such that $(1-\epsilon) f(S) \leq h(S) \leq (1+\epsilon) f(S)$ for all subsets S. Horel & Singer (2016) show that the greedy algorithm achieves a $(1 - 1/e - O(\delta))$ approximation ratio when $\epsilon = \frac{\delta}{k}$. Furthermore, this bound is tight: given a $1/k^{1-\beta}$-approximately submodular function, the greedy algorithm no longer provides a constant factor approximation guarantee.

Elemental Curvature and Total Primal Curvature Wang et al. (2016) analyze the approximation ratio of the greedy algorithm on maximizing non-submodular functions under cardinality constraints. Their bound is $1 - \left(1 - \left(\sum_{i=1}^{k-1} \alpha^i\right)^{-1}\right)^k$ based on the *elemental curvature* with $\alpha = \max_{S \subseteq X, i,j \in X} \frac{f(i|S \cup \{j\})}{f(i|S)}$, and α^i is the i^{th} power of α. Smith & Thai (2017) generalize this definition to the *total primal curvature*.

Algorithm 2: SEMIGRAD for BP maximization

1: **Input:** f, g, constraint set $\mathcal{C}$ and an initial set X_0
2: **Output:** An approximation solution $\hat{X}$.
3: **Initialize:** $i \leftarrow 0$.
4: **repeat**
5: $\quad$ pick a semigradient g_i at X_i of g
6: $\quad X_{i+1} \in \operatorname{argmax}_{X \in \mathcal{C}} f(X) + g_i(X)$;
$\quad$ `//` $\frac{1-e^{-\kappa_f}}{\kappa_f}$`-Approximately solved by Alg 1`
7: $\quad i \leftarrow i + 1$
8: **until** we have converged ($X_i = X_{i-1}$)
9: Return $\hat{X} \leftarrow X_i$

Supermodular Degree Feige and et al. (Feige & Izsak, 2013) introduce a parameter, the supermodular degree, for solving the welfare maximization problem. Feldman and et al. (Feldman & Izsak, 2014b;a) use this concept to analyze monotone set function maximization under a p-extendable system constraint with guarantees. A supermodular degree of one element $u \in V$ by a set function h is defined as the cardinality of the set $\mathcal{D}_h^+(u) = \{v \in V | \exists_{S \subseteq V} h(u|S+v) > h(u|S)\}$, containing all elements whose existence in a set might increase the marginal contribution of u. The supermodular degree of h is $\mathcal{D}_h^+ = \max_{u \in V} |D_h^+(u)|$. A set system $(V, \mathcal{I})$ is called p-extendable (Feldman & Izsak, 2014b;a) if for every two subsets $T \subseteq S \in \mathcal{I}$ and element $u \notin T$ for which $T \cup u \in \mathcal{I}$, there exists a subset $Y \subseteq S \setminus T$ of cardinality at most p for which $S \setminus Y + u \in \mathcal{I}$, which is a generalization of the intersection of p matroids. They offer a greedy algorithm for maximizing a monotonic non-decreasing set function h subject to a p-extendable system with an guarantee of $\frac{1}{p(\mathcal{D}_h^+ + 1) + 1}$ and time complexity polynomial in n and $2^{\mathcal{D}_h^+}$ (Feldman & Izsak, 2014b;a), where $n = |V|$. But again, $\mathcal{D}_h^+$ can not be calculated in polynomial time in general unlike our curvatures. Moreover, if we consider a simple supermodular function $g(X) = |X|^{1+\alpha}$ where α is a small positive number. Then $\mathcal{D}_h^+ = n-1$ since all elements have supermodular interactions. Therefore, the time complexity of their algorithm is polynomial in 2^{n-1} and their bound is $\frac{1}{pn+1}$, while our algorithm requires at most n^2 quires with a performance guarantee of $\frac{1-\log(n)\kappa^g}{p}$ where $\kappa^g = 1 - \frac{1}{n^{1+\alpha}-(n-1)^{1+\alpha}}$. When α is small, our bound is around n times better than theirs; e.g., $n = 10$, $p = 5$, $\alpha = 0.05$, ours is around $\frac{1}{7.61}$ while theirs is $\frac{1}{51}$.

Discussion The above results are both useful and complementary with our analyses below for BP-decomposable functions, thus broadening our understanding of settings where the greedy and semi-gradient algorithms offer a guarantee. We say our analysis is complementary in a sense the following example demonstrates. Should a given function h have a BP decomposition $h = f + g$, then it is easy, given oracle access to both f and g, to compute curvatures and establish bounds. On the other hand, if we do not know h's BP decomposition, or if h does not admit a BP decomposition (Lemma 3.2), then we would need to resort, for example, to the submodularity ratio and generalized curvature bounds of Bian et al. (2017).

2. Approximation Algorithms for BP Maximization

GREEDMAX (Alg. 1) The simplest and most well known algorithm for approximate constrained non-monotone submodular maximization is the greedy algorithm (Nemhauser et al., 1978b). We show that this also works boundedly well for BP maximization when the functions are not both fully curved ($\kappa_f \leq 1, \kappa^g < 1$). At each step, a feasible element with the highest gain with respect to the current set is chosen and added to the set. Finally, if no more elements are feasible, the algorithm returns the greedy set.

SEMIGRAD (Alg. 2) Akin to convex functions, supermodular functions have tight modular lower bounds. These bounds are related to the subdifferential $\partial_g(Y)$ of the supermodular set function g at a set $Y \subseteq V$, which is defined (Fujishige, 2005)[2] as:

$$\partial_g(Y) = \{y \in \mathbb{R}^n : g(X) - y(X) \geq g(Y) - y(Y) \text{ for all } X \subseteq V\} \quad (2)$$

It is possible, moreover, to provide specific semigradients (Iyer & Bilmes, 2012b; Iyer et al., 2013b) that define the following two modular lower bounds:

$$m_{g,X,1}(Y) \triangleq g(X) - \sum_{j \in X \setminus Y} g(j|X \setminus j) + \sum_{j \in Y \setminus X} g(j|\emptyset), \quad (3)$$

$$m_{g,X,2}(Y) \triangleq g(X) - \sum_{j \in X \setminus Y} g(j|V \setminus j) + \sum_{j \in Y \setminus X} g(j|X). \quad (4)$$

Then $m_{g,X,1}(Y), m_{g,X,2}(Y) \leq g(Y), \forall Y \subseteq V$ and $m_{g,X,1}(X) = m_{g,X,2}(X) = g(X)$. Removing constants yields normalized non-negative (since g is monotone) modular functions for g_i in Alg. 2.

Having formally defined the modular lower bound of g, we are ready to discuss how to apply this machinery to BP maximization. SEMIGRAD consists of two stages. In the first stage, it is initialized by an arbitrary set (e.g., $\emptyset$, V, or the solution of GREEDMAX). In the second stage, SEMIGRAD replaces g by its modular lower bound, and solves the resulting problem using GREEDYMAX. The algorithm repeatedly updates the set and calculates an updated modular lower bound until convergence.

Since SEMIGRAD does no worse than the arbitrary initial set, we may start with the solution of GREEDMAX and show that SEMIGRAD is always no worse than GREEDMAX. Interestingly, we obtain the same bounds for SEMIGRAD even if we start with the empty set (Theorems 3.8 and 3.9) despite that they may behave quite differently empirically and yield different solutions (Section 5).

3. Analysis of Approximation Algorithms for BP Maximization

We next analyze the performance of two algorithms GREEDMAX (Alg. 1) and SemiGrad(Alg. 2) under a cardinality constraint and under p matroid constraints. First, we claim that BP maximization is hard and can not be approximately solved to any factor in polynomial time in general.

Lemma 3.1. *(usul, 2016) There exists an instance of a BP maximization problem that can not be approximately solved to any positive factor in polynomial time.*

Proof. For completeness, Appendix A offers a detailed proof based on (usul, 2016). □

It is also important to realize that not all monotone functions are BP-decomposable, as the following demonstrates.

Lemma 3.2. *There exists a monotonic non-decreasing set function h that is not BP decomposable.*

Proof. See Appendix B. □

[2](Fujishige, 2005) defines the subdifferential of a submodular set function. The subdifferential definition for a supermodular set function takes the same form, although instances of supermodular subdifferentials (e.g., Eq. (3)-(4)) take a form different than instances of submodular subdifferentials.

3.1. Supermodular Curvature

Although BP maximization is therefore not possible in general, we show next that we can get worst-case lower bounds using curvature whenever the functions in question indeed have limited curvature.

The (total) curvature of a submodular function f is defined as $\kappa_f = 1 - \min_{v \in V} \frac{f(v|V\setminus\{v\})}{f(v)}$ (Conforti & Cornuejols, 1984). Note that $0 \le \kappa_f \le 1$ since $0 \le f(v|V \setminus \{v\}) \le f(v)$ and if $\kappa_f = 0$ then f is modular. We observed that for any monotonically non-decreasing supermodular function $g(X)$, the dual submodular function (Fujishige, 2005) $g(V) - g(V \setminus X)$ is always monotonically non-decreasing and submodular. Hence, the definition of submodular curvature can be naturally extended to supermodular functions g:

Definition 3.3. T*he* supermodular curvature *of a non-negative monotone nondecreasing supermodular function is defined as* $\kappa^g = \kappa_{g(V)-g(V\setminus X)} = 1 - \min_{v\in V} \frac{g(v)}{g(v|V\setminus\{v\})}$.

For clarity of notation, we use a superscript for supermodular curvature and a subscript for submodular curvature, which also indicates the duality between the two. In fact, for supermodular curvature, we can recover the submodular curvature.

Corollary 3.3.1. $\kappa_f = \kappa^{f(V)-f(V\setminus X)}$.

The dual form also implies similar properties, e.g., we have that $0 \le \kappa^g \le 1$ and if $\kappa^g = 0$ then g is modular. In both cases, a form of curvature indicates the degree of submodularity or supermodularity. If $\kappa_f = 1$ (or $\kappa^g = 1$), we say that f (or g) is fully curved. Intuitively, a submodular function is very (or fully) curved if there is a context B and element v at which the gain is close to (or equal to) zero ($f(v|B) \approx 0$), whereas a supermodular function is very (or fully) curved if there is an element v whose valuation is close to (or equal to) zero ($g(v) \approx 0$). We can calculate both submodular and supermodular curvature easily in linear time. Hence, given a BP decomposition of $h = f + g$, we can easily calculate both curvatures, and the corresponding bounds, with only oracle access to f and g.

Proposition 3.4. *Calculating* κ_f *or* κ^g *requires at most* $2|V| + 1$ *oracle queries of* f *or* g.

The steepness (Il'ev, 2001; Sviridenko et al., 2015) of a monotone nonincreasing supermodular function g' is defined as $s = 1 - \min_{v\in V} \frac{g'(v|V\setminus\{v\})}{g'(v|\emptyset)}$. Here, the numerator and denominator are both negative and g need not be normalized. Steepness has a similar mathematical form to the submodular curvature of a nondecreasing submodular function f, i.e., $\kappa_f = 1 - \min_{v\in V} \frac{f(v|V\setminus\{v\})}{f(v|\emptyset)}$, but is distinct from the supermodular curvature. Steepness may be used to offer a bound for the minimization of such nonincreasing supermodular functions (Sviridenko et al., 2015), whereas we in the present work are interested in maximizing nondecreasing BP (and, hence, which also includes supermodular) functions.

3.2. Theoretical Guarantees for GreedMax

3.2.1. Cardinality constraints

In this section, we provide a lower bound for Greedy maximization of a BP function under a cardinality constraint, inspired by the proof in (Conforti & Cornuejols, 1984) where they focus only on submodular functions.

Lemma 3.5. GreedMax *is guaranteed to obtain a solution* $\hat{X}$ *such that*

$$h(\hat{X}) \ge \frac{1}{\kappa_f}\left[1 - \left(1 - \frac{(1-\kappa^g)\kappa_f}{k}\right)^k\right] h(X^*) \qquad (5)$$

where $X^* \in \operatorname{argmax}_{|X|\le k} h(X)$, $h(X) = f(X) + g(X)$, κ_f *is the curvature of submodular* f *and* κ^g *is the curvature of supermodular* g.

Proof. See Appendix D. □

Theorem 3.6. Theoretical guarantee in the cardinality constrained case. GreedMax *is guaranteed to obtain a solution* $\hat{X}$ *such that*

$$h(\hat{X}) \ge \frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] h(X^*) \qquad (6)$$

where $X^* \in \operatorname{argmax}_{|X|\le k} h(X)$, $h(X) = f(X) + g(X)$, κ_f *is the curvature of submodular* f *and* κ^g *is the curvature of supermodular* g.

Proof. This follows Lemma 3.5 and uses the inequality $(1 - \frac{a}{k})^k \le e^{-a}$ for all $a \ge 0$ and $k \ge 1$. □

Theorem 3.6 gives a lower bound of GreedMax in terms of the submodular curvature κ_f and the supermodular curvature κ^g. We notice that this bound immediately generalizes known results and provides a new one.

1. $\kappa_f = 0$, $\kappa^g = 0$, $h(\hat{X}) = h(X^*)$. In this case, the BP problem reduces to modular maximization under a cardinality constraint, which is solved exactly by the greedy algorithm.
2. $\kappa_f > 0$, $\kappa^g = 0$, $h(\hat{X}) \ge \frac{1}{\kappa_f}\left[1 - e^{-\kappa_f}\right] h(X^*)$. In this case, BP problem reduces to submodular maximization under a cardinality constraint, and with the same $\frac{1}{\kappa_f}\left[1 - e^{-\kappa_f}\right]$ guarantee for the greedy algorithm (Conforti & Cornuejols, 1984).
3. If we take $\kappa_f \to 0$, we get $1 - \kappa^g$, which is a new curvature-based bound for monotone supermodular maximization subject to a cardinality constraint.

4. $\kappa^g = 1$, $h(\hat{X}) \geq 0$ which means, in the general fully curved case for g, this offers no theoretical guarantee for constrained BP or supermodular maximization, consistent with (usul, 2016) and Lemma 3.1.

Another much weaker bound using the two curvatures can also be achieved using greedy on a surrogate objective and that takes the form $h(\hat{X}) \geq \frac{1-\kappa^g}{\kappa_f}\left[1 - e^{-\kappa_f}\right] h(X^*)$. This bound is consistently less than or equal to our presented bound (see Appendix E).

3.2.2. Multiple matroid constraints

Matroids are useful combinatorial objects for expressing constraints in discrete problems, and which are made more useful when taking the intersection of the independent sets of $p > 1$ matroids defined on the same ground set (Nemhauser et al., 1978b). In this section, we show that the greedy algorithm on a BP function subject to p matroid independent constraints has a guarantee if g is not fully curved.

Theorem 3.7. Theoretical guarantee in the p matroids case. GreedMax *is guaranteed to obtain a solution $\hat{X}$ such that*

$$h(\hat{X}) \geq \frac{1-\kappa^g}{(1-\kappa^g)\kappa_f + p} h(X^*) \tag{7}$$

where $X^* \in \operatorname{argmax}_{X \in \mathcal{M}_1 \cap \ldots \cap \mathcal{M}_p} h(X)$, $h(X) = f(X) + g(X)$, κ_f is the curvature of submodular f and κ^g is the curvature of supermodular g.

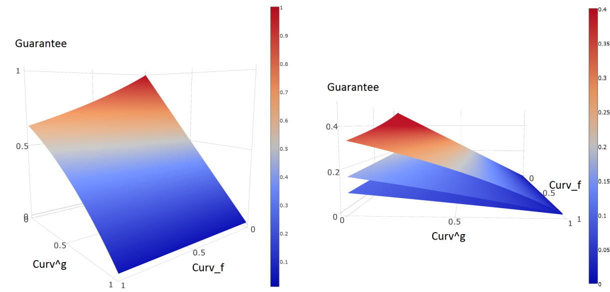
Proof. See Appendix F. □

Theorem 3.7 gives a theoretical lower bound of GreedMax in terms of submodular curvature κ_f and supermodular curvature κ^g for the p matroid constraints case. Like in the cardinality case, this bound also generalizes known results and yields a new one.

1. $\kappa_f = 0$, $\kappa^g = 0$, $h(\hat{X}) \geq \frac{1}{p} h(X^*)$. In this case, the BP problem reduces to modular maximization under p matroid constraints (Conforti & Cornuejols, 1984).
2. $\kappa_f > 0$, $\kappa^g = 0$, $h(\hat{X}) \geq \frac{1}{p+\kappa_f} h(X^*)$. In this case, the BP problem reduces to submodular maximization under p matroid constraints (Conforti & Cornuejols, 1984).
3. If we take $\kappa_f \to 0$, we get $(1-\kappa^g)/p$, which is a new curvature-based bound for monotone supermodular maximization subject to a p matroid constraints.
4. $\kappa^g = 1$, $h(\hat{X}) \geq 0$ which means that, in general, there is no theoretical guarantee for constrained BP or supermodular maximization.

3.3. Theoretical guarantee of SemiGrad

In this section, we show a perhaps interesting result that SemiGrad achieves the same bounds as GreedMax even



(a) Cardinality (b) Multiple matroids

Figure 1. Guarantees of GreedMax for two constraint types. The x and y axes are κ_f and κ^g, respectively, and the z axis is the guarantee. In (b), from top to bottom, the surfaces represent $p = 2, 5, 10$.

if we initialize SemiGrad with $\emptyset$ and even though the two algorithms can produce quite different solutions (as demonstrated in Section 5).

Theorem 3.8. SemiGrad *initialized with the empty set is guaranteed to obtain a solution $\hat{X}$ for the cardinality constrained case such that*

$$h(\hat{X}) \geq \frac{1}{\kappa_f}\left[1 - e^{-(1-\kappa^g)\kappa_f}\right] h(X^*) \tag{8}$$

where $X^* \in \operatorname{argmax}_{|X| \leq k} h(X)$, $h(X) = f(X) + g(X)$, & κ_f (resp. κ^g) is the curvature of f (resp. g).

Proof. See Appendix G. □

Theorem 3.9. SemiGrad *initialized with the empty set is guaranteed to obtain a solution $\hat{X}$, feasible for the p matroid constraints, such that*

$$h(\hat{X}) \geq \frac{1-\kappa^g}{(1-\kappa^g)\kappa_f + p} h(X^*) \tag{9}$$

where $X^* \in \operatorname{argmax}_{X \in \mathcal{M}_1 \cap \ldots \cap \mathcal{M}_p} h(X)$, $h = f + g$, & κ_f (resp. κ^g) is the curvature of f (resp. g).

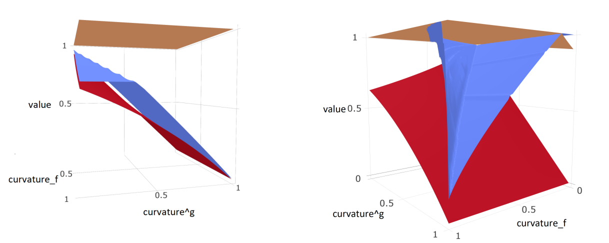
Proof. See Appendix H. □

All the above guarantees are plotted in Figure 1 (in the matroid case for $p = 2$, 5, or 10 matroids).

4. Hardness

We next show that the curvature κ^g limits the polynomial time approximability of BP maximization.

Theorem 4.1. Hardness for cardinality constrained case. *For all $0 \leq \beta \leq 1$, there exists an instance of a BP function $h = f + g$ with supermodular curvature $\kappa^g = \beta$*



(a) (b)

Figure 2. Empirical test of our guarantee. The upper and middle surface indicate the performance of SEMIGRAD and GREEDMAX respectively, and the lower surface is the theoretical worst case guarantee. (a) and (b) are two sets of experiments.

such that no poly-time algorithm solving Problem 1 with a cardinality constraint can achieve an approximation factor better than $1 - \kappa^g + \epsilon$, for any $\epsilon > 0$.

Proof. See Appendix I. □

For the p matroid constraints case, Hazan et al. (2006) studied the complexity of approximating p-set packing which is defined as follows: given a family of sets over a certain domain, find the maximum number of disjoint sets, which is actually a special case of finding the maximum intersection of p matroids. They claim that this problem cannot be efficiently approximated to a factor better than $O(\ln p/p)$ unless P = NP. We generalize their result to BP maximization.

Theorem 4.2. Hardness for p matroids constraint case. *For all $0 \leq \beta \leq 1$, there exists an instance of a BP function $h = f + g$ with supermodular curvature $\kappa^g = \beta$ such that no poly-time algorithm can achieve an approximation factor better than $(1 - \kappa^g)O(\frac{\ln p}{p})$ unless P=NP.*

Proof. See Appendix J. □

Corollary 4.2.1. *No polynomial algorithm can beat* GREEDMAX *or* SEMIGRAD *by a factor of $\frac{1+\epsilon}{1-e^{-1}}$ for cardinality, or $O(\ln(p))$ for p matroid constraints, unless P=NP.*

5. Computational Experiments

We empirically test our guarantees for BP maximization subject to a cardinality constraint on contrived functions using GREEDMAX and SemiGrad. For the first experiment, we let $|V| = 20$ set the cardinality constraint to $k = 10$, and partition the ground set into $|V_1| = |V_2| = k$, $V_1 \cup V_2 = V$ where $V_1 = \{v_1, v_2, \ldots, v_k\}$. Let $w_i = \frac{1}{\alpha}\left[\left(1 - \frac{\alpha}{k}\right)^i - \left(1 - \frac{\alpha}{k}\right)^{i+1}\right]$ for $i = 1, 2, \ldots, k$. Then we define the submodular and supermodular functions as follows, $f(X) = \left[\frac{k - \alpha|X \cap V_2|}{k}\right] \sum_{\{i: v_i \in X\}} w_i + \frac{|X \cap V_2|}{k}$, $g(X) = |X| - \beta \min(1 + |X \cap V_1|, |X|, k) + \epsilon \max(|X|, |X| + \frac{\beta}{1-\beta}(|X \cap V_2| - k + 1))$ and $h(X) = \lambda f(X) + (1-\lambda) g(X)$ for $0 \leq \alpha, \beta, \lambda \leq 1$ and $\epsilon = 1 \times 10^{-5}$. Immediately, we notice that $\kappa_f = \alpha$ and $\kappa^g = \beta$. In particular, we choose $\alpha, \beta, \lambda = 0, 0.01, 0.02, \ldots, 1$ and for all cases, we normalize $h(X)$ using either exhaustive search so that OPT $= h(X^*) = 1$. Since we are doing a proof-of-concept experiment to verify the guarantee, we are interested in the worst case performance at curvatures κ_f and κ^g. In Figure 2(a), we see that both methods are always above the theoretical worst case guarantee, as expected. Interestingly, SEMIGRAD is doing significantly better than GREEDMAX demonstrating the different behavior of the algorithms, despite their identical guarantee. Moreover, the gap between GREEDMAX and the bound layer is small (the maximum difference is 0.1852), which suggests the guarantee for greedy may be almost tight in this case.

The above example is designed to show the tightness of GREEDMAX and the better potential performance of SEMIGRAD. For a next experiment, we again let $|V| = 20$ and $k = 10$, partition the ground set into $|V_1| = |V_2| = k$, $V_1 \cup V_2 = V$. Let $f(X) = |X \cap V_1|^\alpha$ and $g(X) = \max(0, \frac{|X \cap V_2| - \beta}{1-\beta})$ $0 \leq \alpha, \beta \leq 1$, and normalize h (by exhaustive search) to ensure OPT $= h(X^*) = 1$. Immediately, we notice that the curvature of f is $\kappa_f = 1 - k^\alpha + (k-1)^\alpha$ and the curvature of g is $\kappa^g = \beta$. The objective BP function is $h(X) = f(X) + g(X)$. We see that SemiGrad is again doing better than GREEDMAX in most but not all cases (Figure 2(b)) and both are above their bounds, as they should be.

6. Discussion/Conclusions

We have provided a practical constant factor multiplicative guarantee to the greedy and a semigradient algorithm for the family of non-submodular functions admitting a BP decomposition. Future work will advance settings where computable guarantees are possible for non-submodular functions outside the BP family.

Acknowledgments This work was done in part while author Bilmes was visiting the Simons Institute for the Theory of Computing in Berkeley, CA. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1162606, the National Institutes of Health under award R01GM103544, and by a Google, a Microsoft, a Facebook, and an Intel research award. This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA

References

Bai, W., Bilmes, J., and Noble, W. S. Bipartite matching generalizations for peptide identification in tandem mass spectrometry. In *7th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB)*, ACM SIGBio, Seattle, WA, October 2016. ACM, ACM SIGBio.

Bang-Jensen, J., Gutin, G., and Yeo, A. When the greedy algorithm fails. *Discrete Optimization*, 1(2):121 – 127, 2004. ISSN 1572-5286. doi: https://doi.org/10.1016/j.disopt.2004.03.007. URL http://www.sciencedirect.com/science/article/pii/S1572528604000222.

Bednorz, W. (ed.). *Advances in greedy algorithms*, volume 14. Wienna: I-Tech Education and Publishing KG, 2008.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. *arXiv preprint arXiv:1703.02100*, 2017.

Bilmes, J. and Bai, W. Deep Submodular Functions. *Arxiv*, abs/1701.08939, Jan 2017. URL http://arxiv.org/abs/1701.08939.

Conforti, M. and Cornuejols, G. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.

Cormen, T. H. *Introduction to algorithms*. MIT press, 2009.

Cunningham, W. On submodular function minimization. *Combinatorica*, 5(3):185–192, 1985.

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 International Conference on Machine Learning*, pp. 1057–1064. Omnipress, 2011.

Dietrich, B. L. and Hoffman, A. J. On greedy algorithms, partially ordered sets, and submodular functions. *IBM Journal of Research and Development*, 47(1):25–30, 2003.

Dress, A. W. and Wenzel, W. Valuated matroids: A new look at the greedy algorithm. *Applied Mathematics Letters*, 3(2):33–35, 1990.

Dunstan, F. and Welsh, D. A greedy algorithm for solving a certain class of linear programmes. *Mathematical Programming*, 5(1):338–353, 1973.

Edmonds, J. Submodular functions, Matroids and Certain Polyhedra. *Combinatorial structures and their Applications*, 1970.

Edmonds, J. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971.

Faigle, U. The greedy algorithm for partially ordered sets. *Discrete Mathematics*, 28(2):153–159, 1979.

Feige, U. and Izsak, R. Welfare maximization and the supermodular degree. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 247–256. ACM, 2013.

Feldman, M. and Izsak, R. Constrained monotone function maximization and the supermodular degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pp. 160–175, 2014a. doi: 10.4230/LIPIcs.APPROX-RANDOM.2014.160. URL https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.160.

Feldman, M. and Izsak, R. Constrained monotone function maximization and the supermodular degree. *arXiv preprint arXiv:1407.6328*, 2014b.

Fisher, M., Nemhauser, G., and Wolsey, L. An analysis of approximations for maximizing submodular set functions—ii. *Polyhedral combinatorics*, pp. 73–87, 1978.

Fujishige, S. *Submodular functions and optimization*, volume 58. Elsevier Science, 2005.

Goemans, M. X., Harvey, N. J., Iwata, S., and Mirrokni, V. Approximating submodular functions everywhere. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pp. 535–544. Society for Industrial and Applied Mathematics, 2009.

Hazan, E., Safra, S., and Schwartz, O. On the complexity of approximating k-set packing. *computational complexity*, 15(1):20–39, 2006.

Horel, T. and Singer, Y. Maximization of approximately submodular functions. In *Advances In Neural Information Processing Systems*, pp. 3045–3053, 2016.

Huffman, D. A. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

Il'ev, V. P. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114(1):131–146, 2001.

Iyer, R. and Bilmes, J. Algorithms for approximate minimization of the difference between submodular functions, with applications. *Uncertainty in Artificial Intelligence (UAI)*, 2012a.

Iyer, R. and Bilmes, J. A. Submodular-bregman and the lovász-bregman divergences with applications. In *Advances in Neural Information Processing Systems*, pp. 2933–2941, 2012b.

Iyer, R., Jegelka, S., and Bilmes, J. Curvature and Optimal Algorithms for Learning and Minimizing Submodular Functions . In *Neural Information Processing Society (NIPS)*, 2013a.

Iyer, R., Jegelka, S., and Bilmes, J. Fast semidifferential-based submodular function optimization. In *International Conference on Machine Learning*, pp. 855–863, 2013b.

Iyer, R. K. and Bilmes, J. A. Algorithms for approximate minimization of the difference between submodular functions, with applications. *CoRR*, abs/1207.0560, 2012c. URL http://arxiv.org/abs/1207.0560.

Jegelka, S. and Bilmes, J. A. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, 2011.

Karp, B. and Kung, H.-T. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254. ACM, 2000.

Kempner, Y., Levit, V. E., and Muchnik, I. Quasi-concave functions and greedy algorithms. In *Greedy Algorithms*. InTech, 2008.

Kohli, P., Kumar, M. P., and Torr, P. H. P^3 & beyond: Move making algorithms for solving higher order functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31 (9):1645–1656, 2009.

Korte, B., Lovász, L., and Schrader, R. *Greedoids*, volume 4. Springer Science & Business Media, 2012.

Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 2–10. ACM, 2006.

Kruskal, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

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 class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520. Association for Computational Linguistics, 2011a.

Lin, H. and Bilmes, J. Word alignment via submodular maximization over matroids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 170–175. Association for Computational Linguistics, 2011b.

Lin, H. and Bilmes, J. Learning mixtures of submodular shells with application to document summarization. In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

Lin, H., Bilmes, J., and Xie, S. Graph-based submodular selection for extractive summarization. In *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*, Merano, Italy, December 2009.

Liu, Y., Chong, E. K., and Pezeshki, A. Improved bounds for the greedy strategy in optimization problems with curvatures. *arXiv preprint arXiv:1705.04195*, 2017.

Melamed, I. D. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.

Narasimhan, M. and Bilmes, J. A submodular-supermodular procedure with applications to discriminative structure learning. In *Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, Scotland, July 2005. Morgan Kaufmann Publishers.

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

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, 1978b.

Prim, R. C. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957.

Ruiz, R. and Stützle, T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, 2007.

Sarpatwar, K. K., Schieber, B., and Shachnai, H. Interleaved algorithms for constrained submodular function maximization. *arXiv preprint arXiv:1705.06319*, 2017.

Smith, J. D. and Thai, M. T. Breaking the bonds of submodularity: Empirical estimation of approximation ratios for monotone non-submodular greedy maximization. *arXiv preprint arXiv:1702.07002*, 2017.

Sviridenko, M., Vondrák, J., and Ward, J. Tight bounds for submodular and supermodular optimization with bounded curvature. *CoRR*, abs/1311.4728, 2013. URL `http://arxiv.org/abs/1311.4728`.

Sviridenko, M., Vondrák, J., and Ward, J. Optimal approximation for submodular and supermodular optimization with bounded curvature. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1134–1148. Society for Industrial and Applied Mathematics, 2015.

Svitkina, Z. and Fleischer, L. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

Topkis, D. M. *Supermodularity and complementarity*. Princeton university press, 2011.

usul. Maximizing a monotone supermodular function subject to a cardinality constraint. Theoretical Computer Science Stack Exchange, 2016. URL `https://cstheory.stackexchange.com/q/33967`.

Wang, Z., Moran, B., Wang, X., and Pan, Q. Approximation for maximizing monotone non-decreasing set functions with a greedy method. *Journal of Combinatorial Optimization*, 31(1): 29–43, 2016.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning (ICML)*, Lille, France, 2015.

Wolsey, L. A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. A greedy algorithm for aligning dna sequences. *Journal of Computational biology*, 7(1-2):203–214, 2000.