On maximizing a monotone $k$-submodular function subject to a matroid constraint

Shinsaku Sakaue*

August 12, 2016

Abstract

A $k$-submodular function is an extension of a submodular function in that its input is given by $k$ disjoint subsets instead of a single subset. For unconstrained nonnegative $k$-submodular maximization, Ward and Živný proposed a constant-factor approximation algorithm, which was improved by the recent work of Iwata, Tanigawa and Yoshida presenting a $1/2$-approximation algorithm. Iwata et al. also provided a $k/(2k-1)$-approximation algorithm for monotone $k$-submodular maximization and proved that its approximation ratio is asymptotically tight. More recently, Ohsaka and Yoshida proposed constant-factor algorithms for monotone $k$-submodular maximization with several size constraints. However, while submodular maximization with various constraints has been extensively studied, no approximation algorithm has been developed for constrained $k$-submodular maximization, except for the case of size constraints.

In this paper, we prove that a greedy algorithm outputs a $1/2$-approximate solution for monotone $k$-submodular maximization with a matroid constraint. The algorithm runs in $O(M |E|(MO+kEO))$ time, where $M$ is the size of a maximal optimal solution, $|E|$ is the size of the ground set, and $MO$, $EO$ represent the time for the membership oracle of the matroid and the evaluation oracle of the $k$-submodular function, respectively.

1 Introduction

Let $E$ be a finite set and $2^E$ be the family of all subsets in $E$. A function $f : 2^E \rightarrow \mathbb{R}$ is called submodular if it satisfies

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$$

for all pairs of $X, Y \in 2^E$. It is well known that the following diminishing return property characterizes the submodular function:

$$f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y)$$

for any $X \subseteq Y$ and $e \in E \setminus Y$. The diminishing return property often appears in practice, and so various problems can be formulated as submodular function maximization (e.g., sensor placement [13, 14], feature selection [11], and document summarization [16]). Unfortunately, submodular function maximization is known to be NP-hard. Therefore,
approximation algorithms that can run in polynomial time have been extensively studied for submodular function maximization, some of which consider various constraints (e.g., [13] [14] [17] [19]).

Recently, Huber and Kolmogorov [8] proposed k-submodular functions, which express the submodularity on choosing k disjoint sets of elements, instead of a single set. More precisely, let \((k + 1)^E := \{(X_1, \ldots, X_k) \mid X_i \subseteq E \ (i = 1, \ldots, k), \ X_i \cap X_j = \emptyset \ (i \neq j)\}\). Then, a function \(f : (k + 1)^E \rightarrow \mathbb{R}\) is called k-submodular if, for any \(x = (X_1, \ldots, X_k)\) and \(y = (Y_1, \ldots, Y_k)\) with \(X_i \subseteq Y_i\) for \(i = 1, \ldots, k\). It is known that k-submodular functions arise as relaxation of NP-hard problems. k-submodular functions also appear in many applications. Therefore, the k-submodular function is recently a popular subject of study [6, 7]. If \(k = 1\), the above definition is equivalent to that of submodular functions. If \(k = 2\), the k-submodular function is equivalent to the so-called bisubmodular function, for which maximization algorithms have been widely studied [9, 20]. For unconstrained nonnegative k-submodular maximization, Ward and Živný [20] proposed a max\(\{1/3, 1/(1 + a)\}\)-approximation algorithm, where \(a = \max\{1, \sqrt{(k - 1)/4}\}\). Iwata et al. [10] improved the approximation ratio to 1/2. They also proposed a \(k/(2k - 1)\)-approximation algorithm for monotone k-submodular maximization, and proved that, for any \(\varepsilon > 0\), a \((k + 1)/2k + \varepsilon)\)-approximation algorithm for maximizing monotone k-submodular functions requires exponentially many queries. This means their approximation ratio is asymptotically tight. More recently, Ohsaka and Yoshida [18] proposed a 1/2-approximation algorithm for monotone k-submodular maximization with a total size constraint (i.e., \(\bigcup_{i \in \{1, \ldots, k\}} X_i \leq N\) for a nonnegative integer \(N\)) and a 1/3-approximation algorithm for that with individual size constraints (i.e., \(|X_i| \leq N_i\) for \(i = 1, \ldots, k\) with associated nonnegative integers \(N_1, \ldots, N_k\)

In this paper, we prove that 1/2-approximation can be achieved for monotone k-submodular maximization with a matroid constraint. This approximation ratio is asymptotically tight due to the aforementioned hardness result by Iwata et al. [10]. Given \(\mathcal{F} \subseteq 2^E\), we say a system \((E, \mathcal{F})\) is matroid if the following holds:

\((\text{M1})\) \(\emptyset \in \mathcal{F}\),

\((\text{M2})\) If \(A \subseteq B \in \mathcal{F}\) then \(A \in \mathcal{F}\),

\((\text{M3})\) If \(A, B \in \mathcal{F}\) and \(|A| < |B|\) then there exists \(e \in B \setminus A\) such that \(A \cup \{e\} \in \mathcal{F}\).

The elements of \(\mathcal{F}\) are called independent, and we say \(A \in \mathcal{F}\) is maximal if no \(B \in \mathcal{F}\) satisfies \(A \subsetneq B\). Matroids include various systems; the total size constraint can be written as a special case of a matroid constraint. For example, the following systems \((E, \mathcal{F})\) are matroids:
(a) $E$ is a finite set, and $\mathcal{F} := \{F \subseteq E \mid |F| \leq N\}$ where $N$ is a nonnegative integer.

(b) $E$ is the set of columns of a matrix over some field, and $\mathcal{F} := \{F \subseteq E \mid \text{The columns in } F \text{ are linearly independent over the field}\}$.

(c) $E$ is the set of edges of a undirected graph $G$ with a vertex set $V$, and $\mathcal{F} := \{F \subseteq E \mid \text{The graph } (V, F) \text{ is a forest}\}$.

(d) $E$ is a finite set partitioned into $\ell$ sets $E_1, \ldots, E_\ell$ with associated nonnegative integers $N_1, \ldots, N_\ell$, and $\mathcal{F} := \{F \subseteq E \mid |F \cap E_i| \leq N_i \text{ for } i = 1, \ldots, \ell\}$.

The total size constraint corresponds to (a), which is called a uniform matroid. Since submodular functions and matroids are capable of modeling various problems, approximation algorithms for submodular function maximization (i.e., $k = 1$) with a matroid constraint have been extensively studied [2] [3] [4] [5] [15]. However, to the best of our knowledge, no approximation algorithm has been studied for $k$-submodular maximization with a matroid constraint. Therefore, we show that a greedy algorithm provides a $1/2$-approximate solution for the following monotone $k$-submodular maximization with a matroid constraint:

\[
\text{maximize } f(x) \quad \text{subject to } \bigcup_{\ell \in \{1, \ldots, k\}} X_\ell \in \mathcal{F},
\]

where $x = (X_1, \ldots, X_k)$. We also show that our algorithm incurs $O(M|E|(\text{MO} + k\text{EO}))$ computation cost, where $M$ is the size of a maximal optimal solution, and MO, EO represent the time for the membership oracle of the matroid and the evaluation oracle of the $k$-submodular function, respectively. We see in Section 2 that all maximal optimal solutions for problem (1) have equal size, which we denote by $M$ throughout this paper.

The rest of this paper is organized as follows. Section 2 reviews some basics of $k$-submodular functions and matroids. Section 3 discusses a greedy algorithm for problem (1) and proves the $1/2$-approximation. We conclude this paper in Section 4.

### 2 Preliminaries

We elucidate some properties of a $k$-submodular function $f$ where $k \in \mathbb{N}$. Let $[k] := \{1, 2, \ldots, k\}$. For $x = (X_1, \ldots, X_k)$ and $y = (Y_1, \ldots, Y_k)$ in $(k + 1)^E$, we define a partial order $\preceq$ such that $x \preceq y$ if $X_i \subseteq Y_i$ for all $i \in [k]$. For $x, y \in (k + 1)^E$ satisfying $x \preceq y$, we use $x < y$ if $X_i \subsetneq Y_i$ holds for some $i \in [k]$. We also define

$$\Delta_{e,i} f(x) := f(X_1, \ldots, X_{i-1}, \{e\}, X_i, \ldots, X_k) - f(X_1, \ldots, X_k)$$

for $x \in (k + 1)^E$, $e \notin \bigcup_{\ell \in [k]} X_\ell$ and $i \in [k]$, which is a marginal gain when adding $e \in E$ to the $i$-th set of $x \in (k + 1)^E$. It is not hard to see that the $k$-submodularity implies the orthant submodularity [20]:

$$\Delta_{e,i} f(x) \geq \Delta_{e,i} f(y)$$

for any $x, y \in (k + 1)^E$ with $x \preceq y$, $e \notin \bigcup_{j \in [k]} Y_j$, and $i \in [k]$, and the pairwise monotonicity:

$$\Delta_{e,i} f(x) + \Delta_{e,j} f(x) \geq 0$$

for any $x \in (k + 1)^E$, $e \notin \bigcup_{\ell \in [k]} X_\ell$, and $i, j \in [k]$ with $i \neq j$. Actually, these properties characterize $k$-submodular functions:

**Theorem 1** (Ward and Živný [20]). A function $f : (k + 1)^E \to \mathbb{R}$ is $k$-submodular if and only if $f$ is orthant submodular and pairwise monotone.
For notational ease, we identify \((k + 1)^E\) with \(\{0, 1, \ldots, k\}^E\), that is, we associate \((X_1, \ldots, X_k) \in (k + 1)^E\) with \(x \in \{0, 1, \ldots, k\}^E\) by \(X_i = \{e \in E \mid x(e) = i\}\) for \(i \in [k]\). We sometimes abuse the notation, and simply write \(x = (X_1, \ldots, X_k)\) by regarding a vector \(x\) as disjoint \(k\) subsets of \(E\). For \(x \in \{0, 1, \ldots, k\}^E\), we define \(\text{supp}(x) := \{e \in E \mid x(e) \neq 0\}\); the size of \(x\) can be written as \(|\text{supp}(x)|\). Let \(0\) be the zero vector in \(\{0, 1, \ldots, k\}^E\). In what follows, we assume that the monotone \(k\)-submodular function \(f\) in problem (1) satisfies \(f(0) = 0\) without loss of generality; if \(f(0) \neq 0\), we redefine \(f(x) := f(x) - f(0)\) where \(x \in (k + 1)^E\).

We now turn to some properties of matroid \((E,F)\). An independent set \(A \in F\) is called a bases if it is a maximal independent set. We denote the set of all bases by \(B\). It is known that each element in \(B\) has the same size (see, e.g., [12, Theorem 13.5]); the size is denoted by \(M\) throughout this paper. Thus, we have the following lemma for the size of the maximal optimal solutions for problem (1).

**Lemma 1.** The size of any maximal optimal solution for problem (1) is \(M\).

**Proof.** Assume there is a maximal optimal solution \(o\) such that \(|\text{supp}(o)| < M\). Let \(x \in (k + 1)^E\) be an arbitrary vector such that \(\text{supp}(x) \in B\). Then, by (M3), there exists \(e \in \text{supp}(x)\) such that \(\text{supp}(o) \cup \{e\} \in F\). Since \(f\) is monotone, by assigning arbitrary \(i \in [k]\) to \(o(e)\), we get \(\Delta_{e,i}f(o) \geq 0\); more precisely, \(\Delta_{e,i}f(o) = 0\) since \(o\) is an optimal solution. This contradicts to the assumption that \(o\) is a maximal optimal solution. \(\square\)

We also introduce the following lemma for later use.

**Lemma 2.** Suppose \(A \in F\) and \(B \in B\) satisfy \(A \subseteq B\). Then, for any \(e \notin A\) satisfying \(A \cup \{e\} \in F\), there exists \(e' \in B \setminus A\) such that \(\{B \setminus \{e'\}\} \cup \{e\} \in B\).

**Proof.** If \(|B| - |A| = 1\), by defining \(e' = B \setminus A\), we get \(\{B \setminus \{e'\}\} \cup \{e\} = A \cup \{e\} \in F\). Since \(|A \cup \{e\}| = |B|\), we have \(\{B \setminus \{e'\}\} \cup \{e\} \in B\).

If \(|B| - |A| \geq 2\), then \(|A \cup \{e\}| < |B|\). Thus, by applying (M3) iteratively, we can obtain \(|B| - |A| - 1\) elements \(e_1, \ldots, e_{|B| - |A| - 1} \in B \setminus \{A \cup \{e\}\}\) such that

\[
\{A \cup \{e\}\} \cup \{e_1\} \cup \cdots \cup \{e_{|B| - |A| - 1}\} \in B.
\]

Therefore, defining \(e' = B \setminus \{A \cup \{e_1\} \cup \cdots \cup \{e_{|B| - |A| - 1}\}\}\), we get

\[
\{B \setminus \{e'\}\} \cup \{e\} = \{A \cup \{e\}\} \cup \{e_1\} \cup \cdots \cup \{e_{|B| - |A| - 1}\} \in B.
\]

This completes the proof. \(\square\)

### 3 Maximizing a monotone \(k\)-submodular function with a matroid constraint

We present a greedy algorithm for problem (1): it runs in \(O(M|E|(MO + kEO))\) time where \(MO\) and \(EO\) stand for the time for the membership oracle of matroid and the evaluation oracle of \(k\)-submodular function, respectively. We then prove that the greedy algorithm outputs a \(1/2\)-approximate solution for problem (1). In summary, this section proves the following theorem:

**Theorem 2.** For problem (1), a \(1/2\)-approximate solution can be obtained in \(O(M|E|(MO + kEO))\) time.
Algorithm 3.1 A greedy algorithm for $k$-submodular maximization with a matroid constraint

**Input:** a monotone $k$-submodular function $f : (k + 1)^E \rightarrow \mathbb{R}$ and a matroid $(E, \mathcal{F})$.

**Output:** a vector $s$ satisfying $\text{supp}(s) \in \mathcal{B}$.

1: $s \leftarrow \emptyset$.
2: for $j = 1$ to $M$ do
3: \quad $e_{\text{last}} \leftarrow \emptyset$, Value $\leftarrow 0$.
4: \quad for each $e \in E \setminus \text{supp}(s)$ such that $\text{supp}(s) \cup \{e\} \in \mathcal{F}$ do
5: \quad \quad $i \leftarrow \arg \max_{i \in [k]} \Delta_{e,i}f(s)$
6: \quad \quad if $\Delta_{e,i}f(s) \geq \text{Value}$ then
7: \quad \quad \quad $s(e_{\text{last}}) \leftarrow 0$ unless $e_{\text{last}} = \emptyset$.
8: \quad \quad \quad $s(e) \leftarrow i$.
9: \quad \quad \quad $e_{\text{last}} \leftarrow e$ and Value $\leftarrow \Delta_{e,i}f(s)$.
10: \quad end if
11: end for
12: end for
13: return $s$.

3.1 Greedy algorithm and its complexity analysis

We consider applying Algorithm 3.1 to problem \[1\]. First, we make a remark on using Algorithm 3.1 in practice. In Step 2, the algorithm requires the value of $M$, the size of a maximal independent set. However, in practice, we need not calculate the value of $M$ beforehand. Instead, we continue the iteration while there exists $e \in E \setminus \text{supp}(s)$ satisfying $\text{supp}(s) \cup \{e\} \in \mathcal{F}$, which we check in Step 4. We can confirm that this modification does not change the output as follows. As long as $|\text{supp}(s)| < M$, exactly one element is added to $\text{supp}(s)$ at each iteration due to the monotonicity and (M3), and, if $|\text{supp}(s)| = M$, the iteration stops since $\text{supp}(s)$ is a maximal independent set. Algorithm 3.1 is described using $M$ to make it easy to understand the subsequent discussions. Note that, defining $s^{(j)}$ as the solution obtained after the $j$-th iteration, we have $|\text{supp}(s^{(j)})| = j$ for $j \in [M]$.

We now examine the time complexity of Algorithm 3.1. Let $E\text{O}$ be the time for the evaluation oracle of the $k$-submodular function $f$, and MO be the time for the membership oracle of the matroid $(E, \mathcal{F})$. At the $j$-th iteration, the membership oracle is used at most $|E|$ times in Step 4, and the evaluation oracle is used at most $k|E|$ times in Step 5. Thus, the time complexity of Algorithm 3.1 is given by $O(M|E|(MO + kE))$.

3.2 Proof for 1/2-approximation

We now prove that Algorithm 3.1 gives a 1/2-approximate solution for problem \[1\]. To prove this, we define a sequence of vectors $o^{(0)}, o^{(1)}, \ldots, o^{(M)}$ as in \[10\], \[18\], \[20\].

Let $(e^{(j)}, i^{(j)})$ be the pair chosen greedily at the $j$-th iteration, and $s^{(j)}$ be the solution after the $j$-th iteration; we let $s = s^{(M)}$, the output of Algorithm 3.1. We define $s^{(0)} := \emptyset$ and let $o$ be a maximal optimal solution. In what follows, we show how to construct a sequence of vectors $o^{(0)} = o, o^{(1)}, \ldots, o^{(M-1)}, o^{(M)} = s$ satisfying the following:

(2) $s^{(j)} \preceq o^{(j)}$ if $j = 0, 1, \ldots, M - 1$, and $s^{(j)} = o^{(j)} = s$ if $j = M$.

(3) $O^{(j)} \in \mathcal{B}$ for $j = 0, 1, \ldots, M$.

More specifically, we see how to obtain $o^{(j)}$ from $o^{(j-1)}$ satisfying (2) and (3). Note that $s^{(0)} = \emptyset$ and $o^{(0)} = o$ satisfy (2) and (3). We define $S^{(j)} := \text{supp}(s^{(j)}), O^{(j)} := \text{supp}(o^{(j)})$
for each $j \in [M]$.

We now describe how to obtain $o^{(j)}$ from $o^{(j-1)}$, assuming that $o^{(j-1)}$ satisfies

$$s^{(j-1)} \prec o^{(j-1)},$$

and $O^{(j-1)} \in \mathcal{B}$.

Since $s^{(j-1)} \prec o^{(j-1)}$ means $S^{(j-1)} \subseteq O^{(j-1)}$, and $e^{(j)}$ is chosen to satisfy $S^{(j-1)} \cup \{e^{(j)}\} \in \mathcal{F}$, we see from Lemma 2 that there exists $e' \in O^{(j-1)} \setminus S^{(j-1)}$ satisfying $\{o^{(j-1)}\} \cup \{e^{(j)}\} \in \mathcal{B}$. We let $o^{(j)} = e'$ and define $o^{(j-1/2)}$ as the vector obtained by assigning 0 to the $o^{(j)}$-th element of $o^{(j-1)}$. We then define $o^{(j)}$ as the vector obtained from $o^{(j-1/2)}$ by assigning $i^{(j)}$ to the $e^{(j)}$-th element. The vector thus constructed, $o^{(j)}$, satisfies

$$O^{(j)} = \{O^{(j-1)} \setminus \{o^{(j)}\}\} \cup \{e^{(j)}\} \in \mathcal{B}.$$ 

Furthermore, since $o^{(j-1/2)}$ satisfies

$$s^{(j-1)} \prec o^{(j-1/2)},$$

we have the following property for $o^{(j)}$:

$$s^{(j)} \prec o^{(j)} \text{ if } j = 1, \ldots, M - 1, \text{ and } s^{(j)} = o^{(j)} = s \text{ if } j = M,$$

where the strictness of the inclusion for $j \in [M - 1]$ can be easily confirmed from $|S^{(j)}| = j < M = |O^{(j)}|$. Thus, applying the above discussion for $j = 1, \ldots, M$ iteratively, we see from (4) and (5) that the obtained sequence of vectors $o^{(0)}, o^{(1)}, \ldots, o^{(M)}$ satisfies (2) and (4).

We now prove the following inequality for $j \in [M]$:

$$f(s^{(j)}) - f(s^{(j-1)}) \geq f(o^{(j-1)}) - f(o^{(j)}).$$

Since $S^{(j-1)} \cup \{o^{(j)}\} \subseteq O^{(j-1)} \in \mathcal{B}$ holds for each $j \in [M]$, we get the following inclusion from (M2):

$$S^{(j-1)} \cup \{o^{(j)}\} \in \mathcal{F}$$

for any $j \in [M]$. Therefore, for the pair $(e^{(j)}, i^{(j)})$, which is chosen greedily, we have

$$\Delta_{e^{(j)}, i^{(j)}} f(s^{(j-1)}) \geq \Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(s^{(j-1)}).$$

Furthermore, since $s^{(j-1)} \prec o^{(j-1/2)}$ holds, orthant submodularity implies

$$\Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(s^{(j-1)}) \leq \Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(o^{(j-1/2)}).$$

Using (4) and (5), we get

$$f(s^{(j)}) - f(s^{(j-1)}) \geq \Delta_{e^{(j)}, i^{(j)}} f(s^{(j-1)})$$

$$\geq \Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(s^{(j-1)})$$

$$\geq \Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(o^{(j-1/2)})$$

$$\geq \Delta_{o^{(j)}, o^{(j-1)}, (o^{(j)})} f(o^{(j-1/2)}) - \Delta_{e^{(j)}, i^{(j)}} f(o^{(j-1/2)})$$

$$= f(o^{(j-1)}) - f(o^{(j)}),$$

where the third inequality comes from the monotonicity, i.e., $\Delta_{e^{(j)}, i^{(j)}} f(o^{(j-1/2)}) \geq 0$.

By (5), we have

$$f(o) - f(s) = \sum_{j=1}^{M} (f(o^{(j-1)}) - f(o^{(j)})) \leq \sum_{j=1}^{M} (f(s^{(j)}) - f(s^{(j-1)})) = f(s) - f(0) = f(s),$$

which means $f(s) \geq f(o)/2$. 
4 Conclusions

We proved that a $1/2$-approximate solution can be obtained for monotone $k$-submodular maximization with a matroid constraint via a greedy algorithm. Our approach follows the techniques shown in [10, 18, 20]. The proved approximation ratio is asymptotically tight due to the hardness result shown in [10]. We also showed that the proposed algorithm incurs $O(M|E|(MO + kEO))$ computation cost.

References

[1] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz. A tight linear time $(1/2)$-approximation for unconstrained submodular maximization. *SIAM J. Comput.*, 44(5):1384–1402, 2015.

[2] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[3] Y. Filmus and J. Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Foundations of Computer Science, 2012 IEEE 53rd Annual Symposium on*, pages 659–668. IEEE, 2012.

[4] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions-II. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

[5] D. Golovin and A. Krause. Adaptive submodular optimization under matroid constraints. *arXiv preprint arXiv:1101.4450*, 2011.

[6] I. Gridchyn and V. Kolmogorov. Potts model, parametric maxflow and $k$-submodular functions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2320–2327, 2013.

[7] H. Hirai and Y. Iwamasa. On $k$-submodular relaxation. *arXiv preprint arXiv:1504.07830*, 2015.

[8] A. Huber and V. Kolmogorov. Towards minimizing $k$-submodular functions. In *Proceedings of 2nd International Symposium on Combinatorial Optimization*, pages 451–462. Springer, 2012.

[9] S. Iwata, S. Tanigawa, and Y. Yoshida. Bisubmodular function maximization and extensions. Technical report, Technical Report METR 2013-16, The University of Tokyo, 2013.

[10] S. Iwata, S. Tanigawa, and Y. Yoshida. Improved approximation algorithms for $k$-submodular function maximization. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 404–413. SIAM, 2016.

[11] C.-W. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Oper. Res.*, 43(4):684–691, 1995.

[12] B. Korte and J. Vygen. *Combinatorial Optimization*, volume 2. Springer, 2012.

[13] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *J. Mach. Learn. Res.*, 9(Dec):2761–2801, 2008.
[14] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9(Feb):235–284, 2008.

[15] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM J. Discrete. Math.*, 23(4):2053–2078, 2010.

[16] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics, 2010.

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

[18] N. Ohsaka and Y. Yoshida. Monotone $k$-submodular function maximization with size constraints. In *Advances in Neural Information Processing Systems*, pages 694–702, 2015.

[19] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.

[20] J. Ward and S. Živný. Maximizing $k$-submodular functions and beyond. *arXiv preprint arXiv:1409.1399*, 2014.