Regularized nonmonotone submodular maximization

Cheng Lu, Wenguo Yang and Suixiang Gao

School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, People’s Republic of China

ABSTRACT
In this paper, we present a thorough study of the regularized submodular maximization problem, in which the objective \( f := g - \ell \) can be expressed as the difference between a submodular function and a modular function. This problem has drawn much attention in recent years. While existing works focuses on the case of \( g \) being monotone, we investigate the problem with a nonmonotone \( g \). The main technique we use is to introduce a distorted objective function, which varies weights of the submodular component \( g \) and the modular component \( \ell \) during the iterations of the algorithm. By combining the weighting technique and measured continuous greedy algorithm, we present an algorithm for the matroid-constrained problem, which has a provable approximation guarantee. In the cardinality-constrained case, we utilize random greedy algorithm and sampling technique together with the weighting technique to design two efficient algorithms. Moreover, we consider the unconstrained problem and propose a much simpler and faster algorithm compared with the algorithms for solving the problem with a cardinality constraint.

1. Introduction
Given a set \( \Omega \) consisting of \( n \) elements, we say a set function \( f: 2^\Omega \to \mathbb{R} \) is submodular if it satisfies diminishing returns property, i.e. \( f(A \cup \{e\}) - f(A) \leq f(B \cup \{e\}) - f(B) \) for every \( A \subseteq B \subset \Omega \) and \( e \in \Omega \setminus B \). A few examples of submodular function include rank functions of matroids, cut functions of graphs and digraphs, entropy functions and covering functions. In the last decade, submodular optimization problem has been studied extensively due to its wide applications in viral marketing [1], machine learning [2, 3], economic theory [4, 5], etc. Unlike minimization of submodular functions which can be done in polynomial time [6–8], submodular maximization problems are usually \( NP \)-hard for many classes of submodular functions, such as weighted coverage [9] or mutual information [10].
In general, a submodular maximization problem can be formulated as follows:

$$\max\{f(A) : A \in \mathcal{F} \subseteq 2^\Omega\},$$  \hspace{1cm} (1)

where \(f : 2^\Omega \rightarrow \mathbb{R}_+\) is a submodular function, and \(\mathcal{F}\) is a collection of feasible sets. The goal of Problem (1) is to select a set of elements \(A\) obeying constraint such that the utility \(f(A)\) is maximized. Clearly, Problem (1) doesn’t take the cost of selecting elements into consideration. However, from a problem-modelling perspective, in many real-world applications such as team formation and profit maximization [11], we should aim for a balance between the utility and the cost of picking a certain set of elements. Formally, when the cost of elements is considered, a generalization of Problem (1) can be formulated as follows:

$$\max\{g(A) - \ell(A) : A \in \mathcal{F}\},$$  \hspace{1cm} (2)

where \(g : 2^\Omega \rightarrow \mathbb{R}_+\) is a submodular function and \(\ell : 2^\Omega \rightarrow \mathbb{R}_+\) is a modular function. In Problem (2), we use \(g\) to encode the utility of sets and \(\ell\) to represent the cost. This formulation also finds its applications in machine learning [12], in which the modular function \(\ell\) represents a penalty or a regularizer term to alleviate over-fitting. In this paper, we refer to Problem (2) as the regularized submodular maximization problem.

1.1. Related work

Submodular maximization: Various studies have been devoted to submodular maximization problems. A classical result [13] showed that a simple greedy algorithm can provide a \((1 - 1/e)\)-approximation for the cardinality-constrained submodular maximization problem (i.e. \(\mathcal{F} = \{A \subseteq \Omega : |A| \leq k\}\)) with a monotone objective. When the constraint is independent sets of a general matroid, the greedy algorithm can return a 1/2-approximate solution [14]. On the hardness side, Nemhauser and Wolsey [15] proved that the \(1 - 1/e\) approximation ratio is tight for maximizing a monotone submodular function subject to a cardinality constraint. A natural question arises: does there exist a polynomial-time algorithm which can produce a \((1 - 1/e)\)-approximate solution to Problem (1) with a general matroid constraint? Calinescu et al. [16] gave an affirmative answer to this question. They proposed the continuous greedy algorithm which can provide a \(1 - 1/e\) approximation guarantee. To be specific, their algorithm consists of two main components, i.e. a relaxation solver and a rounding procedure. First, they approximately solve a relaxation problem of Problem (1), i.e.

$$\max\{F(x) : x \in \mathcal{P} \subseteq [0, 1]^\Omega\},$$

where \(F\) is the multilinear extension of submodular function \(f : 2^\Omega \rightarrow \mathbb{R}_+\), and \(\mathcal{P}\) is the matroid polytope corresponding to matroid \(\mathcal{M} = (\Omega, \mathcal{I})\). Then, they utilize...
a rounding procedure called pipage rounding to obtain an integral solution, which
does not lose anything in the objective compared with the fractional solution.

The results mentioned above are all about submodular maximization with
a monotone objective. Another line of research focuses on the case in which
the objective function is nonmonotone. Buchbinder et al. [17] proposed an $1/e$
approximation algorithm called random greedy for the cardinality-constrained
problem. In addition, they also presented a much more involved algorithm which
Can achieve an improved approximation ratio of $1/e + 0.004$. On the hardness
side, Gharan and Vondrák [18] proved that there is no algorithm which can
achieve an approximation ratio better than 0.491 for the problem using a polynomial
number of value oracle queries. By modifying the continuous greedy algorithm [16], Feldman et al. [19] proposed an algorithm called measured continuous greedy for the maximization problem with an arbitrary matroid
constraint. They showed that the measured continuous greedy algorithm achieves
an approximation ratio of $1/e - o(1)$. On the inapproximability side, Gharan
and Vondrák [18] also proved that there is no algorithm which can achieve an
approximation ratio better than 0.478 when the constraint is a partition matroid
using a polynomial number of value oracle queries. The unconstrained maxi-
mization problem (i.e. $\mathcal{F} = \mathcal{2}^\Omega$) has also been studied. Buchbinder et al. [20]
proposed a randomized algorithm called double greedy for solving such prob-
lem. They proved that the algorithm achieves an approximation ratio of $1/2$. On
the hardness side, Feige et al. [21] proved that no polynomial-time algorithm for
the unconstrained problem can have an approximation ratio of $1/2 + \epsilon$ for any
constant $\epsilon > 0$ in the value oracle model.

**Regularized submodular maximization:** Recently, much attention has been
attracted to the regularized submodular maximization problem, i.e. Problem (2).
We notice that by the definition of submodularity, the objective function $f :=
g - \ell$ of Problem (2) is still submodular. The only difference between Problem (1)
and Problem (2) is that the objective $f = g - \ell$ of Problem (2) could possibly take
on negative values, while the objective function of Problem (1) is nonnegative. In
fact, Feige et al. [21] have shown that for a submodular function $f$ without any
restrictions, verifying whether the maximum of $f$ is greater than zero is $\mathcal{NP}$-
hard and requires exponentially many queries in the value oracle model. Thus, it
is impossible to design an algorithm which can achieve a multiplicative approxi-
mation for Problem (2).\(^2\) A line of research has shown that in this case we should
consider a weaker notion of approximation, i.e. finding a solution $S \in \mathcal{F}$ such that

$$g(S) - \ell(S) \geq \alpha \cdot g(OPT) - \ell(OPT),$$

where $\alpha$ is a constant in $(0, 1]$ and $OPT$ is an optimal solution.

Sviridenko et al. [22] are the first to study the submodular maximization
problem with an objective which could possibly take on negative values. To be
specific, they considered the problem of maximizing the sum of a nonnegative
submodular function and a normalized modular function, i.e.
\[
\max \{ g(A) + \ell(A) : A \in \mathcal{I} \},
\]
where \( g : 2^\Omega \to \mathbb{R}_+ \) is a monotone submodular function, \( \ell: 2^\Omega \to \mathbb{R} \) is a normalized modular function, and \( \mathcal{I} \) is a collection of independent sets of matroid \( \mathcal{M} = (\Omega, \mathcal{I}) \). They proposed a polynomial-time algorithm which can return a feasible set \( S \) such that \( \mathbb{E}[g(S) + \ell(S)] \geq (1 - 1/e) \cdot g(OPT) + \ell(OPT) - O(\epsilon) \), where \( OPT \) is an optimal solution. The main drawback of their algorithm is that it requires a time-consuming guessing step. Later, Feldman [23] reconsidered this problem and designed a distilled continuous greedy algorithm based on a surrogate objective which can bypass the guessing step. Harshaw et al. [24] adopted the idea of [23] and analogously proposed a distorted discrete greedy algorithm for the cardinality-constrained problem. They showed that such algorithm can return a feasible solution \( S \) with \( g(S) - \ell(S) \geq (1 - e^{-\gamma}) \cdot g(OPT) - \ell(OPT) \), where \( \gamma \) is the submodularity ratio of \( g \). Moreover, they extended their results to the unconstrained case, and presented a much more efficient algorithm which can output a set \( S \) such that \( \mathbb{E}[g(S) - \ell(S)] \geq (1 - e^{-\gamma}) \cdot g(OPT) - \ell(OPT) \). Nikolakaki et al. [11] suggested a new technique for solving Problem (2). Their main idea is to introduce a scaled objective (i.e. \( f := g - \lambda \cdot \ell \)), and the resulting algorithms can produce a solution \( S \) satisfying \( g(S) - \ell(S) \geq 1/2 \cdot g(OPT) - \ell(OPT) \) for both the matroid-constrained and unconstrained problem. Nikolakaki et al. [11] and Kazemi et al. [12] studied Problem (2) with a cardinality constraint in the online setting, and proposed two similar one-pass streaming algorithms which can return a solution \( S \) such that \( g(S) - \ell(S) \geq (3 - \sqrt{5})/2 - \epsilon) \cdot g(OPT) - \ell(OPT) \). Kazemi et al. [12] also presented a distributed algorithm which can produce a solution \( S \) with \( \mathbb{E}[g(S) - \ell(S)] \geq (1 - \epsilon) \cdot [(1 - 1/e) \cdot g(OPT) - \ell(OPT)] \).

1.2. Our contributions

As mentioned in the previous subsection, existing works on regularized submodular maximization assume that \( g \) is monotone. In this paper, we consider the case in which \( g \) is a nonmonotone submodular function. We design algorithms for solving the matroid-constrained, cardinality-constrained and unconstrained problem, and analyse the performance guarantee of those algorithms. Our results are summarized in Table 1. For comparison, Table 2 shows existing results about the case of \( g \) being monotone.³

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we give the necessary preliminaries. We study the regularized nonmonotone maximization problem with a matroid constraint in Section 3 and propose a continuous greedy algorithm
for solving it. In Section 4, we present two fast algorithms for the cardinality-constrained problem. In Section 5, we study the unconstrained problem and design a fast randomized algorithm. We conclude this paper in Section 6.

2. Preliminaries

2.1. Set functions

Given a set \( A \) and an element \( e \), we use \( A + e \) and \( A - e \) as shorthands for the expression \( A \cup \{e\} \) and \( A \setminus \{e\} \), respectively. Let \( \Omega \) be a set of size \( n \). A set function \( f : 2^\Omega \to \mathbb{R} \) is submodular iff \( f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \) for any \( A, B \subseteq \Omega \). Submodularity can equivalently be characterized in terms of marginal gains, which is defined as \( f(e|A) := f(A + e) - f(A) \). Then, \( f \) is submodular iff \( f(e|A) \geq f(e|B) \) for any \( A \subseteq B \subseteq \Omega \) and \( e \in \Omega \setminus B \). We denote the marginal gain of a set \( B \) to a set \( A \) with respect to \( f \) by \( f(B|A) := f(A \cup B) - f(A) \). We say that \( f \) is monotone nondecreasing iff \( f(A) \leq f(B) \) for any \( A \subseteq B \subseteq \Omega \). Similarly, \( f \) is monotone nonincreasing iff \( f(A) \geq f(B) \) for any \( A \subseteq B \subseteq \Omega \). A function \( f : 2^\Omega \to \mathbb{R} \) is said to be normalized if \( f(\emptyset) = 0 \). As a special case of submodular functions, \( f : 2^\Omega \to \mathbb{R} \) is a modular function iff \( f(A) + f(B) = f(A \cup B) + f(A \cap B) \) for any \( A, B \subseteq \Omega \).

Based on the definitions above, it’s easy to see that a normalized set function \( \ell : 2^\Omega \to \mathbb{R} \) is modular iff there exists a vector \( \mathbf{d} \in \mathbb{R}^\Omega \) such that \( \ell(A) = \sum_{e \in A} d_e \) for every \( A \subseteq \Omega \). Thus, in this paper, we abuse notation and identify the normalized modular function \( \ell \) with the vector \( \mathbf{l} = (\ell_e)_{e \in \Omega} \in \mathbb{R}^\Omega \) which it corresponds to. Then, \( \ell(A) = \sum_{e \in A} \ell_e \) for every \( A \subseteq \Omega \).

2.2. Operations on vectors

Given two vectors \( \mathbf{x}, \mathbf{y} \in [0, 1]^\Omega \), we write \( \mathbf{x} \preceq \mathbf{y} \) if \( x_e \leq y_e \) for every element \( e \in \Omega \). We use \( \mathbf{x} \lor \mathbf{y}, \mathbf{x} \land \mathbf{y} \) and \( \mathbf{x} \circ \mathbf{y} \) to denote the coordinate-wise maximum,
minimum and multiplication of \( x \) and \( y \), respectively. Specifically, \((x \lor y)_e := \max\{x_e, y_e\}, (x \land y)_e := \min\{x_e, y_e\}\) and \((x \circ y)_e := x_e \cdot y_e\), for every \( e \in \Omega \).

### 2.3. Multilinear extension

Given a vector \( x \in [0, 1]^\Omega \), let \( R_x \) denote a random subset of \( \Omega \) containing every element \( e \in \Omega \) independently with probability \( x_e \). Then, the **multilinear extension** \( F : [0, 1]^\Omega \rightarrow \mathbb{R} \) of \( f : 2^\Omega \rightarrow \mathbb{R} \) is defined as

\[
F(x) := \mathbb{E}_{R_x}[f(R_x)] = \sum_{S \subseteq \Omega} f(S) \prod_{e \in S} x_e \prod_{e \in \Omega - S} (1 - x_e).
\]

If we denote by \( 1_S \) the characteristic vector of set \( S \subseteq \Omega \), then it holds that \( F(1_S) = f(S) \). Thus, function \( F \) is indeed an extension of \( f \).

By the definition of multilinear extension, it’s easy to verify that for every \( e \in \Omega \) and \( x \in [0, 1]^\Omega \),

\[
(1 - x_e) \frac{\partial F(x)}{\partial x_e} = F(x \lor 1_{\{e\}}) - F(x) = \mathbb{E}_{R_x}[f(e|R_x)],
\]

and

\[
\frac{\partial F(x)}{\partial x_e} = F(x \lor 1_{\{e\}}) - F(x \land 1_{\Omega - \{e\}}) = \mathbb{E}_{R_x}[f(e|R_x - e)].
\]

In addition, for every \( e, e' \in \Omega \) and \( x \in [0, 1]^\Omega \), it holds that

\[
\frac{\partial^2 F(x)}{\partial x_e \partial x'_e} = \mathbb{E}_{R_x}[f(e'|R_x + e - e') - f(e'|R_x - e - e')].
\]

We now consider the multilinear extension of normalized modular functions. Suppose that \( \ell : 2^\Omega \rightarrow \mathbb{R} \) is a normalized modular function, then its multilinear extension is \( L(x) = \mathbb{E}_{R_x}[\ell(R_x)] = \sum_{e \in \Omega} \ell_e \cdot x_e = \langle l, x \rangle \) for every \( x \in [0, 1]^\Omega \).

### 2.4. Lovász extension

Given a vector \( x \in [0, 1]^\Omega \) and a scalar \( \lambda \in [0, 1] \), let \( T_\lambda(x) := \{ e \in \Omega : x_e \geq \lambda \} \) be the set of elements in \( \Omega \) whose coordinate in \( x \) is at least \( \lambda \). Then, the **Lovász extension** \( \hat{f} : [0, 1]^\Omega \rightarrow \mathbb{R} \) of a submodular function \( f : 2^\Omega \rightarrow \mathbb{R} \) is defined as

\[
\hat{f}(x) := \mathbb{E}_{\lambda \sim U[0,1]}[f(T_\lambda(x))] = \int_0^1 f(T_\lambda(x)) \, d\lambda,
\]

where \( U[0,1] \) represents a uniform distribution on the interval \([0, 1]\). In this paper, we will use the Lovász extension to lower bound the multilinear extension via the following lemma.

**Lemma 2.1 (Lemma A.4 in [25]):** Let \( F(x) \) and \( \hat{f}(x) \) be the multilinear and Lovász extensions, respectively, of a submodular function \( f : 2^\Omega \rightarrow \mathbb{R} \). Then, it holds that \( F(x) \geq \hat{f}(x) \) for every \( x \in [0, 1]^\Omega \).
2.5. Polytopes

A polytope $P \subseteq [0,1]^{\Omega}$ is said to be down-monotone if $x \in P$ and $0 \leq y \leq x$ imply $y \in P$. A polytope $P$ is solvable if there is an oracle for optimizing normalized modular functions over $P$, i.e. for solving $\max \{ \langle I, x \rangle : x \in P \}$ for an arbitrary vector $I \in \mathbb{R}^{\Omega}$.

Given a matroid $\mathcal{M} = (\Omega, \mathcal{I})$, the corresponding matroid polytope $P(\mathcal{M})$ is

$$P(\mathcal{M}) := \text{conv}\{1_I : I \in \mathcal{I}\} = \left\{ x \geq 0 : \sum_{e \in S} x_e \leq r(\mathcal{M}(S)), \forall S \subseteq \Omega \right\},$$

where $r$ is the rank function of matroid $\mathcal{M}$. According to the definition, matroid polytope $P(\mathcal{M})$ is down-monotone and solvable.

2.6. Extra assumption

In this paper, we study the regularized nonmonotone submodular maximization problem, i.e. Problem (2) with $g$ being nonmonotone. Here, $g$ is nonmonotone means that $g$ is neither monotone nondecreasing nor monotone nonincreasing. This assumption is reasonable due to the following facts: When $g$ is monotone nondecreasing, the corresponding Problem (2) has been studied by a vast literature [11, 12, 22–24]. When $g$ is monotone nonincreasing, $\emptyset$ is an optimal solution to the problem.

Because $g$ is neither monotone nondecreasing nor monotone nonincreasing, it holds that $\max_{e \in \Omega} g(e|\emptyset) > 0$ and $\min_{e \in \Omega} g(e|\Omega - e) < 0$. Hence, we have

$$M := \max_{e \in \Omega} \{ \max_{e \in \Omega} g(e|\emptyset), -\min_{e \in \Omega} g(e|\Omega - e) \} > 0.$$

Note that given a function $g : 2^{\Omega} \to \mathbb{R}$, parameter $M$ is linear-time computable.

3. Matroid-constrained problem

In this section, we study the regularized nonmonotone submodular maximization problem under a matroid constraint, i.e.

$$\max\{g(A) - \ell(A) : A \in \mathcal{I}\},$$

where $g : 2^{\Omega} \to \mathbb{R}_+$ is a nonmonotone submodular function, $\ell : 2^{\Omega} \to \mathbb{R}_+$ is a normalized modular function, and $\mathcal{I}$ is a collection of independent sets of matroid $\mathcal{M} = (\Omega, \mathcal{I})$. Similar to the matroid-constrained submodular maximization, the algorithm for solving Problem (4) also consists of two steps. First,
we consider the relaxation of Problem (4), i.e.
\[
\max\{G(x) - L(x) : x \in P \subseteq [0, 1]^\Omega\},
\]
where \(G\) and \(L\) are the multilinear extensions of \(g\) and \(\ell\) respectively, and \(P\) is a down-monotone and solvable polytope. We propose a modified measured continuous greedy algorithm for solving this relaxation problem. Second, due to the submodularity of \(g - \ell\), we may use the pipage rounding procedure \([16, 25]\) to obtain an integral solution to Problem (4).

### 3.1. Relaxation solver

The algorithm for solving Problem (5) is presented in Algorithm 1, which is a combination of the measured continuous greedy algorithm proposed by Feldman et al. \([19]\), and the distorted objective introduced by Feldman \([23]\). Note that while Feldman \([23]\) chose \((1 + \delta)^{1-\delta} - 1\) as the coefficient in their distorted objective, we set the coefficient as \((1 - \delta)^{1-\delta} - 1\), which is more appropriate for the problem we study. According to the settings of Algorithm 1, we have \(\delta \leq \min\{1/2, \epsilon/n^2\}\). In what follows, we analyse the performance guarantee of Algorithm 1.

**Algorithm 1: Distorted Measured Continuous Greedy**

**Input:** function \(g\) and \(\ell\), polytope \(P\), error \(\epsilon \in (0, 1)\)

**Output:** approximate fractional solution \(y(1)\)

1. \(y(0) \leftarrow 1\emptyset, t \leftarrow 0, \delta \leftarrow 1/[2 + n^2/\epsilon]\);
2. while \(t < 1\) do
3.   foreach \(e \in \Omega\) do
4.     Let \(w_e(t)\) be an estimate for \(\mathbb{E}[g(e|R_{y(t)})]\) obtained by averaging the value of the expression within this expectation for
5.     \[r = \frac{(2n^2/\epsilon^2) \ln(2n/(\epsilon \delta))}{\delta^2}\] independent samples of \(R_{y(t)}\);
6.   \(z(t) \leftarrow \arg\max\{\langle v, (1 - \delta)^{1-\delta} - 1 w(t) - 1 \rangle : v \in P\}\);
7.   \(y(t + \delta) \leftarrow y(t) + \delta \cdot z(t) \circ (1_\Omega - y(t))\);
8. return \(y(1)\).

#### 3.1.1. Feasibility of solution

Let \(T\) be the set of times considered by Algorithm 1, i.e. \(T := \{i\delta : 0 \leq i < 1/\delta, i \in \mathbb{Z}\}\). First, we prove that each coordinate of \(y(t)\) has a nontrivial upper bound during the iteration.

**Lemma 3.1:** For every \(t \in T \cup \{1\}\) and every \(e \in \Omega\), it holds that \(0 \leq y_e(t) \leq 1 - (1 - \delta)^{t/\delta}\).
Proof: We prove this lemma by induction on $t$.

Basis step: Since $y(0) = 1_{\emptyset}$, the inequalities hold for $t = 0$.

Induction step: Assume that $0 \leq y_e(t) \leq 1 - (1 - \delta)^{t/\delta}$ is valid for some $t \in T$.
Now consider time $t + \delta$. According to Algorithm 1, we have $y_e(t + \delta) = y_e(t) + \delta z_e(t)(1 - y_e(t))$. Due to $z(t) \in P \subseteq [0, 1]^2$, $y_e(t + \delta) \geq 0$ holds.
Meanwhile, by induction assumption, we have $y_e(t + \delta) \leq y_e(t) + \delta(1 - y_e(t)) \leq 1 - (1 - \delta)^{t/\delta + 1}$.

The proof is completed.

Next, we prove that the solution Algorithm 1 produces is feasible.

Corollary 3.2: $y(1) \in P$.

Proof: According to Algorithm 1, we have $y(1) = \sum_{t \in T} \delta \cdot z(t) \circ (1_\Omega - y(t))$. By Lemma 3.1, it holds that $1_{\emptyset} \leq 1 \Omega - y(t) \leq 1_\Omega$ for every $t \in T$, which implies $1_{\emptyset} \leq z(t) \circ (1_\Omega - y(t)) \leq z(t)$. Since $P$ is a down-monotone polytope and $z \in P$, we deduce that $z(t) \circ (1_\Omega - y(t)) \in P$ for every $t \in T$. Thus, $y(1) = \sum_{t \in T} \delta \cdot z(t) \circ (1_\Omega - y(t))$ is a convex combination of vectors in $P$. Then, the convexity of $P$ implies $y(1) \in P$.

3.1.2. Good estimator
Line 4 of Algorithm 1 uses $r$ random samples of $R_{y(t)}$ to estimate $\mathbb{E}[g(e|R_{y(t)})]$. We now prove that it occurs with a low probability that any of the estimates made by Algorithm 1 has a significant error. To do this, we will need the following lemma.

Lemma 3.3 (The symmetric version of Theorem A.1.16 in [26]): Let $X_i (i \in [k])^5$ be mutually independent random variables with $\mathbb{E}[X_i] = 0$ and $|X_i| \leq 1$ for every $i \in [k]$. Let $S = X_1 + X_2 + \cdots + X_k$ and $\alpha$ be a positive number. Then, it holds that $\Pr[|S| > \alpha] \leq 2e^{-\alpha^2/2k}$.

Let $\mathcal{E}(e, t)$ be the event that $|w_e(t) - \mathbb{E}[g(e|R_{y(t)})]| \leq 2\epsilon M/n$ for some $e \in \Omega$ and $t \in T$, and $\mathcal{E}$ be the event $\bigcap_{e \in \Omega} \bigcap_{t \in T} \mathcal{E}(e, t)$, namely, $|w_e(t) - \mathbb{E}[g(e|R_{y(t)})]| \leq 2\epsilon M/n$ for every $e \in \Omega$ and $t \in T$. We show that $\mathcal{E}$ happens with a high probability.

Lemma 3.4: $\Pr[\mathcal{E}] \geq 1 - \epsilon$.

Proof: Consider an arbitrary element $e \in \Omega$ and a time $t \in T$. Let $R_i$ denote the $i$th independent sample of $R_{y(t)}$ used for calculating $w_e(t)$. Then, define random variables $X_i = (g(e|R_i) - \mathbb{E}[g(e|R_{y(t)})])/2M$ for every $i = [r]$.
By the linearity of expectation, we have $\mathbb{E}[X_i] = 0$. Since $g$ is submodular, it holds that $g(e|R_i), \mathbb{E}[g(e|R_{y(t)})] \in [-M, M]$, which implies $X_i \in [-1, 1]$ for every $i = [r]$. Then, by Lemma 3.3, it holds that

$$\Pr \left[ \hat{E} (e, t) = \Pr \left[ \left| w_e(t) - \mathbb{E}[g(e|R_{y(t)})] \right| > 2\epsilon M/n \right] \right] \leq \Pr \left[ \sum_{i=1}^{r} X_i > \epsilon r/n \right] \leq 2e^{-\frac{1}{2} \left( \frac{\epsilon r}{n} \right)^2} = \epsilon \delta / n. $$

By union bound, we have

$$\Pr \left[ \mathcal{E} \right] = \Pr \left[ \bigcup_{e \in \Omega} \bigcup_{t \in T} \mathcal{E} (e, t) \right] \leq \sum_{e \in \Omega} \sum_{t \in T} \Pr \left[ \mathcal{E} (e, t) \right] \leq \epsilon .$$

Thus, $\Pr[\mathcal{E}] = 1 - \Pr[\mathcal{E}] \geq 1 - \epsilon$, which concludes the proof. ■

3.1.3. A technical lemma

The following lemma characterizes the behaviour of multilinear extension of a submodular function within a small neighbour.

**Lemma 3.5:** Given two vectors $y, y' \in [0, 1]^{\Omega}$ such that $|y'_e - y_e| \leq \delta \leq 1$ and a nonnegative submodular function $f: 2^\Omega \to \mathbb{R}_+$ whose multilinear extension is $\hat{F}$. It holds that $|F(y') - F(y) - \langle \nabla F(y), y' - y \rangle| \leq n^2 \delta^2 M$, where $n = |\Omega|$ and $M = \max_{e \in \Omega} f(e|\emptyset), -\min_{e \in \Omega} f(e|\Omega - e) > 0$.

**Proof:** According to Taylor’s theorem, we have

$$F(y') - F(y) = \langle \nabla F(y), y' - y \rangle + \frac{1}{2} \sum_{e, e' \in \Omega} \frac{\partial^2 F(\hat{y})}{\partial x_e \partial x'_e} \cdot (y'_e - y_e)(y'_e - y_e'),$$

where $\hat{y} = y + t(y' - y)$ for some constant $t \in [0, 1]$. Next, we estimate the second order partial derivative of $F$. By the submodularity of $f$ and the definition of $M$, it holds that $f(e'|R_{\hat{y}} + e - e'), f(e'|R_{\hat{y}} - e - e') \in [-M, M]$, which implies

$$\left| \frac{\partial^2 F(\hat{y})}{\partial x_e \partial x'_e} \right| \leq \mathbb{E}_{R_{\hat{y}}} \left[ |f(e'|R_{\hat{y}} + e - e')| + |f(e'|R_{\hat{y}} - e - e')| \right] \leq 2M.$$

Thus, we have

$$|F(y') - F(y) - \langle \nabla F(y), y' - y \rangle| \leq \frac{1}{2} \sum_{e, e' \in \Omega} \left| \frac{\partial^2 F(\hat{y})}{\partial x_e \partial x'_e} \right| \cdot |(y'_e - y_e)(y'_e - y_e')| \leq n^2 \delta^2 M,$$

which concludes the proof. ■
3.1.4. Performance guarantee

Now, we begin to analyse the performance guarantee of Algorithm 1. First, we lower bound the increase of $G$ at each iteration.

**Lemma 3.6:** If event $E$ happens, then for every $t \in T$, it holds that

$$G(y(t + \delta)) - G(y(t)) \geq \delta (w(t), z(t)) - 3\epsilon \delta M,$$

where $M = \max\{\max_{e \in \Omega} g(e|\emptyset), - \min_{e \in \Omega} g(e|\Omega - e)\} > 0$.

**Proof:** By Lemma 3.1 and $z(t) \in \mathcal{P} \subseteq [0, 1]^{\Omega}$, it holds that $|y_e(t + \delta) - y_e(t)| = |\delta z_e(t)(1 - y_e(t))| \leq \delta$ for every $e \in \Omega$. It follows that

$$G(y(t + \delta)) - G(y(t)) \geq (\nabla G(y(t)), y(t + \delta) - y(t)) - n^2\delta^2 M$$

$$= (\nabla G(y(t)), \delta \cdot z(t) \circ (1_\Omega - y(t))) - n^2\delta^2 M$$

$$= \delta \cdot \sum_{e \in \Omega} z_e(t) \cdot \mathbb{E}[g(e|R_y(t))] - n^2\delta^2 M,$$  \hspace{1cm} \hspace{1cm} (6)

where the first inequality follows from Lemma 3.5, the first equality follows from $y(t + \delta) = y(t) + \delta \cdot z(t) \circ (1_\Omega - y(t))$, and the second equality follows from the property of partial derivative of multilinear extension.

When event $E$ happens, $\mathbb{E}[g(e|R_y(t))] \geq w_e(t) - 2\epsilon M/n$ holds for every $e \in \Omega$ and $t \in T$. Plugging this inequality into (6) yields

$$G(y(t + \delta)) - G(y(t)) \geq \delta \cdot \sum_{e \in \Omega} z_e(t) \cdot (w_e(t) - 2\epsilon M/n) - n^2\delta^2 M$$

$$\geq \delta (w(t), z(t)) - 2\epsilon \delta M - n^2\delta^2 M$$

$$\geq \delta (w(t), z(t)) - 3\epsilon \delta M,$$

where the last inequality holds due to $\delta \leq \min\{1/2, \epsilon/n^2\}$. The proof is done. \hspace{1cm} \hspace{1cm} ■

Similarly, we upper bound the increase of $L$ at each iteration.

**Lemma 3.7:** For every $t \in T$, it holds that $L(y(t + \delta)) - L(y(t)) \leq \delta (l, z(t))$.

**Proof:** For every $t \in T$, we have

$$L(y(t + \delta)) - L(y(t)) = (l, y(t + \delta) - y(t))$$

$$= \delta (l, z(t)) - \delta (l, z(t) \circ y(t)) \leq \delta (l, z(t)),$$

where the first equality follows from the property of multilinear extension of a modular function, the second equality follows from $y(t + \delta) = y(t) + \delta \cdot z(t) \circ (1_\Omega - y(t))$, and the inequality holds due to the nonnegativity of $\delta$, $l$, $z(t)$ and $y(t)$. \hspace{1cm} \hspace{1cm} ■
We introduce an auxiliary function which is defined as \( \Phi(t) := (1 - \delta) \frac{1 - t}{\delta} G(y(t)) - L(y(t)) \). Notice that \( \Phi(t) \) varies the relative importance between \( G \) and \( L \) as the algorithm proceeds. Next, we lower bound the increase of \( \Phi \) at each iteration.

**Lemma 3.8:** If event \( E \) happens, then for every \( t \in T \), it holds that

\[
\Phi(t + \delta) - \Phi(t) \geq \delta [1/e \cdot g(OPT) - \ell(OPT)] - 5\epsilon \delta M,
\]

where \( M = \max \{ \max_{e \in \Omega} g(e|\emptyset), - \min_{e \in \Omega} g(e|\Omega - e) \} \) and \( OPT \in \argmax \{ g(A) - \ell(A) : A \subseteq \Omega, 1_A \in \mathcal{P} \} \).

**Proof:** For every \( t \in T \), we have

\[
\Phi(t + \delta) - \Phi(t) = (1 - \delta) \frac{1 - t}{\delta} \left[ G(y(t + \delta)) - G(y(t)) \right] - \left[ L(y(t + \delta)) - L(y(t)) \right] + \delta(1 - \delta) \frac{1 - t}{\delta} G(y(t))
\]

\[
\geq \delta \langle z(t), (1 - \delta) \frac{1 - t}{\delta} w(t) - I \rangle + \delta(1 - \delta) \frac{1 - t}{\delta} G(y(t)) - 3\epsilon \delta M, \tag{7}
\]

where the equality follows from the definition of \( \Phi \), the first inequality follows from Lemma 3.6 and 3.7, and the second inequality follows from \( 0 \leq (1 - \delta) \frac{1 - t}{\delta} \leq 1 \).

According to Line 5 of Algorithm 1, we have

\[
\langle z(t), (1 - \delta) \frac{1 - t}{\delta} w(t) - I \rangle \geq \langle 1_{OPT}, (1 - \delta) \frac{1 - t}{\delta} w(t) - I \rangle = (1 - \delta) \frac{1 - t}{\delta} \sum_{e \in OPT} w_e(t) - \ell(OPT). \tag{8}
\]

Meanwhile, when event \( E \) happens, \( w_e(t) \geq \mathbb{E}[g(e|R_y(t))] - 2\epsilon M/n \) holds for every \( e \in \Omega \) and \( t \in T \). It follows that

\[
\sum_{e \in OPT} w_e(t) \geq \sum_{e \in OPT} (\mathbb{E}[g(e|R_y(t))] - 2\epsilon M/n) \geq \mathbb{E} \left[ g(OPT|R_y(t)) \right] - 2\epsilon M = G(y(t) \lor 1_{OPT}) - G(y(t)) - 2\epsilon M, \tag{9}
\]

where the second inequality follows from the submodularity of \( g \) and \( |OPT| \leq n \), and the equality follows from the definition of multilinear extension. Consider
term $G(y(t) \lor 1_{OPT})$. We have

$$G(y(t) \lor 1_{OPT}) \geq \hat{g}(y(t) \lor 1_{OPT}) = \int_0^1 g(T_{\lambda}(y(t) \lor 1_{OPT})) \, d\lambda \geq \int_1^{1-1/(1-\delta)^{1/\delta}} g(T_{\lambda}(y(t) \lor 1_{OPT})) \, d\lambda = (1-\delta)^{\frac{1}{\delta}} g(OPT),$$

where the first inequality follows from Lemma 2.1, and the last equality holds due to $T_{\lambda}(y(t) \lor 1_{OPT}) = OPT$ for every $\lambda \in [1 - (1 - \delta)^{1/\delta}, 1]$. Combining (8)–(10) yields

$$\langle z(t), (1 - \delta)^{\frac{1}{\delta} - 1} w(t) - l \rangle \geq (1 - \delta)^{\frac{1}{\delta} - 1} g(OPT) - \ell(OPT) - (1 - \delta)^{\frac{1}{\delta} - 1} G(y(t)) - 2\epsilon M. \quad (11)$$

Plugging (11) into (7), we have

$$\Phi(t + \delta) - \Phi(t) \geq \delta \left[(1 - \delta)^{\frac{1}{\delta} - 1} g(OPT) - \ell(OPT)\right] - 5\epsilon \delta M \geq \delta \left[1/e \cdot g(OPT) - \ell(OPT)\right] - 5\epsilon \delta M$$

where the last inequality holds due to $\ln(1 - \delta) \geq -\delta/(1 - \delta)$ for every $\delta \in (0, 1)$.

Now, we can analyse the approximation guarantee of Algorithm 1.

**Lemma 3.9:** If event $E$ happens, it holds that

$$G(y(1)) - L(y(1)) \geq 1/e \cdot g(OPT) - \ell(OPT) - 5\epsilon M,$$

where $M = \max\{\max_{e \in \Omega} g(e|\emptyset), -\min_{e \in \Omega} g(e|\emptyset - e)\}$ and $OPT \in \argmax\{g(A) - \ell(A) : A \subseteq \Omega, 1_A \in \mathcal{P}\}$.

**Proof:** By the definition of $\Phi$, we have $\Phi(1) = G(y(1)) - L(y(1))$ and $\Phi(0) = (1 - \delta)^{1/\delta} g(\emptyset) - \ell(\emptyset) = (1 - \delta)^{1/\delta} g(\emptyset) \geq 0$. It follows that

$$G(y(1)) - L(y(1)) = \Phi(0) + \sum_{i=0}^{1-\delta-1} \left[\Phi(i\delta + \delta) - \Phi(i\delta)\right] \geq \Phi(0) + \delta \cdot \sum_{i=0}^{1-\delta-1} \left[1/e \cdot g(OPT) - \ell(OPT) - 5\epsilon M\right] \geq 1/e \cdot g(OPT) - \ell(OPT) - 5\epsilon M,$$

where the first inequality follows from Lemma 3.8.
At each iteration, Algorithm 1 requires $2nr$ value oracle queries of $g$. Thus, during the $1/δ$ iterations, Algorithm 1 performs $2nr/δ = \tilde{O}(n^5/ε^3)$ value oracle queries in total. Then, based on Lemmas 3.4 and 3.9, we obtain the following theorem.

**Theorem 3.10:** When Algorithm 1 terminates, with a high probability it produces a vector $x \in \mathcal{P}$ such that

$$G(x) - L(x) \geq 1/e \cdot g(OPT) - \ell(OPT) - 5εM,$$

where $M = \max\{\max_{e \in \Omega} g(e|\emptyset), -\min_{e \in \Omega} g(e|\Omega - e)\}$ and $OPT \in \arg\max \{g(A) - \ell(A) : A \subseteq 2^{\Omega}, 1_A \in \mathcal{P}\}$. During the iteration, the total number of value oracle queries is $\tilde{O}(n^5/ε^3)$.

### 3.2. Rounding

Algorithm 1 produces a fractional solution to the relaxation problem (i.e. Problem (5)). Thus, in order to obtain a feasible solution to the original problem (i.e. Problem (4)), we need to round the fractional solution to an integral one. Since the objective $g - \ell$ is submodular and $\mathcal{P}$ is a matroid polytope, there exists rounding techniques such as pipage rounding [16, 25], which can produce an integral solution without losing anything in the objective. Specifically, pipage rounding procedure can output a random independent set $S \in \mathcal{I}$ of matroid $\mathcal{M} = (\Omega, \mathcal{I})$ satisfying

$$\mathbb{E}[g(S) - \ell(S)] \geq G(y(1)) - L(y(1)) \geq 1/e \cdot g(OPT) - \ell(OPT) - 5εM.$$ 

### 4. Cardinality-constrained problem

In this section, we investigate the cardinality-constrained problem, i.e.

$$\max\{g(A) - \ell(A) : |A| \leq k\}. \quad (12)$$

This problem is a special case of the matroid-constrained problem, which indicates that we may apply Algorithm 1 for solving this problem. However, due to the characteristic of cardinality constraint, we can design much more efficient algorithms for Problem (12). In this section, we assume that $k \geq 2$, since the case of $k = 1$ is trivial.

Before presenting our algorithms, we define two auxiliary functions which was introduced by Harshaw et al. [24]. For every $i = 0, 1, \ldots, k$ and every $T \subseteq \Omega$, define

$$\Phi_i(T) := (1 - 1/k)^{k-i} \cdot g(T) - \ell(T).$$

In addition, for every $i = 0, 1, \ldots, k - 1$, $T \subseteq \Omega$ and $e \in \Omega$, define

$$\Psi_i(T, e) := \max\{0, (1 - 1/k)^{k-(i+1)} \cdot g(e|T) - \ell_e\}. $$
4.1. Distorted random greedy

Now, we present our first algorithm which is a combination of the random greedy algorithm [17] and the distorted objective proposed by Harshaw et al. [24]. Thus, we call this algorithm Distorted Random Greedy. At the ith iteration, Algorithm 2 finds top $r(r \leq k)$ elements which make maximum positive distorted marginal contribution (i.e. $(1 - 1/k)^{k-(i+1)} \cdot g(e|S_i) - \ell_e)$ to the current solution $S_i$, and then add a uniformly random element $e_i$ into $S_i$. After $k$ iterations, Algorithm 2 returns a feasible solution. First, we consider the increase of $\Phi$ at each iteration.

\begin{algorithm}[H]
\caption{Distorted Random Greedy}
\begin{algorithmic}[1]
\State \textbf{Input:} function $g$ and $\ell$, cardinality $k$
\State \textbf{Output:} approximate solution $S_k$
\State $S_0 \leftarrow \emptyset$;
\For{$i = 0$ \textbf{to} $k - 1$}
\State $M_i \leftarrow \text{argmax}\{\sum_{e \in B}((1 - 1/k)^{k-(i+1)} \cdot g(e|S_i) - \ell_e) : B \subseteq \Omega, |B| \leq k\}$;
\hspace{1em} // Here, we implicitly assume that when $|M_i| > 0$, $(1 - 1/k)^{k-(i+1)} \cdot g(e|S_i) - \ell_e > 0$ holds for every $e \in M_i$.
\State \hspace{1em} with probability $1 - |M_i|/k$ $S_{i+1} \leftarrow S_i$;
\State \hspace{1em} otherwise Let $e_i$ be a uniformly random element of $M_i$, and set $S_{i+1} \leftarrow S_i + e_i$;
\EndFor
\State \textbf{return} $S_k$.
\end{algorithmic}
\end{algorithm}

Lemma 4.1: At each iteration $(i = 0, 1, \ldots, k - 1)$ of Algorithm 2, it holds that

$$\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i}[\Phi_i(S_i)] \geq \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)}$$

$$\times \mathbb{E}_{S_i}[g(OPT \cup S_i)] - \frac{1}{k} \ell(OPT),$$

where $OPT \in \text{argmax}\{g(A) - \ell(A) : A \subseteq \Omega, |A| \leq k\}$.

Proof: Since $S_{i+1} = S_i$ or $S_{i+1} = S_i + e_i$ when $S_i$ is given, we have

$$\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i]$$

$$= \Pr[S_{i+1} = S_i|S_i] \cdot (\Phi_{i+1}(S_i) - \Phi_i(S_i))$$

$$+ \sum_{e \in M_i} \Pr[S_{i+1} = S_i + e|S_i] \cdot (\Phi_{i+1}(S_i + e) - \Phi_i(S_i)).$$
According to the settings of Algorithm 2, we have \( \Pr[S_{i+1} = S_i | S_i] = 1 - |M_i|/k \) and \( \Pr[S_{i+1} = S_i + e | S_i] = 1/k \) for every \( e \in M_i \). Meanwhile, by the definition of \( \Phi \), it holds that

\[
\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \left( 1 - \frac{1}{k} \right)^{k-(i+1)} [g(S_{i+1}) - g(S_i)] + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} \times g(S_i) - [\ell(S_{i+1}) - \ell(S_i)].
\]

Hence, when \( S_{i+1} = S_i \), then

\[
\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i).
\]

When \( S_{i+1} = S_i + e \) for some \( e \in M_i \), since \( (1 - 1/k)^{k-(i+1)} \cdot g(e|S_i) - \ell_e > 0 \) and \( \ell \) is nonnegative, we have \( e \notin S_i \), which implies

\[
\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(e|S_i) - \ell_e + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i).
\]

Thus, it holds that

\[
\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i] = \left( 1 - \frac{|M_i|}{k} \right) \cdot \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i) + \frac{1}{k} \sum_{e \in M_i} \left[ \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(e|S_i) - \ell_e + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i) \right]
\]

\[
= \frac{1}{k} \sum_{e \in M_i} \left[ \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(e|S_i) - \ell_e \right] + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i) \geq \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(OPT|S_i) - \frac{1}{k} \ell(OPT) + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i) \]

\[
= \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(OPT \cup S_i) - \frac{1}{k} \ell(OPT),
\]

where the first inequality follows from the choice of \( M_i \) and \( |OPT| \leq k \), and the second inequality follows from the submodularity of \( g \). Then, we have

\[
\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i}[\Phi_i(S_i)] = \mathbb{E}_{S_i} \left[ \mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i] \right] \]

\[
\geq \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} \mathbb{E}_{S_i}[g(OPT \cup S_i)] - \frac{1}{k} \ell(OPT),
\]

which completes the proof. \[\blacksquare\]
Next, we lower bound $\mathbb{E}_{S_i}[g(\text{OPT} \cup S_i)]$ in terms of $g(\text{OPT})$. To do this, we will need the following lemma.

**Lemma 4.2 (Lemma 2.2 in [17]):** Let $f: 2^\Omega \to \mathbb{R}_+$ be a submodular function. Denote by $A(p)$ a random subset of $A \subseteq \Omega$ where each element appears with probability at most $p$ (not necessarily independently). Then, $\mathbb{E}_{A(p)}[f(A(p))] \geq (1 - p)f(\emptyset)$.

The key to apply Lemma 4.2 to lower bound $\mathbb{E}_{S_i}[g(\text{OPT} \cup S_i)]$ is to analyse the maximum probability with which every element $e \in \Omega$ belongs to $S_i$.

**Lemma 4.3:** For every $i = 0, 1, \ldots, k - 1$, it holds that $\mathbb{E}_{S_i}[g(\text{OPT} \cup S_i)] \geq (1 - 1/k)^i \cdot g(\text{OPT})$.

**Proof:** Suppose that $e$ is an arbitrary element in $\Omega$. When event $e \notin S_i$ happens, then $e_i \neq e$ implies $e \notin S_{i+1}$. Namely, $\Pr[e \notin S_{i+1}|e \notin S_i] = \Pr[e_i \neq e | e \notin S_i]$.

If $e \notin M_i$, then $\Pr[e_i \neq e | e \notin S_i] = 1$. If $e \in M_i$, then $\Pr[e_i \neq e | e \notin S_i] = 1 - 1/k$. As a result, for any $e \in \Omega$ and any $i = 0, 1, \ldots, k - 1$, it holds that $\Pr[e \notin S_{i+1}|e \notin S_i] \geq 1 - 1/k$. Then, for every $i = 1, 2, \ldots, k - 1$, we have

$$\Pr[e \notin S_i] = \Pr[e \notin S_0, e \notin S_1, \ldots, e \notin S_{i-1}, e \notin S_i]$$

$$= \Pr[e \notin S_0] \cdot \Pr[e \notin S_1|e \notin S_0] \cdots$$

$$\Pr[e \notin S_i|e \notin S_0, e \notin S_1, \ldots, e \notin S_{i-1}]$$

$$= \Pr[e \notin S_0] \cdot \Pr[e \notin S_1|e \notin S_0] \cdots \Pr[e \notin S_i|e \notin S_{i-1}] \geq (1 - 1/k)^i,$$

where the last equality holds since $e \notin S_{j}$ implies $e \notin S_r$ for every $r = 0, 1, \ldots, j - 1$. Thus, $\Pr[e \in S_i] = 1 - \Pr[e \notin S_i] \leq 1 - (1 - 1/k)^i$ for every $e \in \Omega$ and $i \in [k - 1]$.

Define $h: 2^\Omega \to \mathbb{R}_+$ as $h(A) := g(\text{OPT} \cup A)$ for every $A \subseteq \Omega$. Note that $h$ is a submodular function. Thus, by Lemma 4.2, we have $\mathbb{E}_{S_i}[h(S_i)] \geq (1 - 1/k)^i \cdot h(\emptyset)$, which implies $\mathbb{E}_{S_i}[g(\text{OPT} \cup S_i)] \geq (1 - 1/k)^i \cdot g(\text{OPT})$ for every $i \in [k - 1]$. Since $\mathbb{E}_{S_0}[g(\text{OPT} \cup S_0)] = g(\text{OPT})$, the proof is completed.

Using Lemmas 4.1 and 4.3, we can analyse the performance guarantee of Algorithm 2.

**Theorem 4.4:** When Algorithm 2 terminates, it returns a feasible set $S_k$ with

$$\mathbb{E}_{S_k}[g(S_k) - \ell(S_k)] \geq 1/e \cdot g(\text{OPT}) - \ell(\text{OPT}),$$

where $\text{OPT} \in \arg\max\{g(A) - \ell(A) : A \subseteq \Omega, |A| \leq k\}$. During the execution, Algorithm 2 performs $O(kn)$ value oracle queries.
**Proof:** According to Lemmas 4.1 and 4.3, we have

\[
E_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - E_{S_i}[\Phi_i(S_i)] \geq \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-1} g(OPT) - \frac{1}{k} \ell(OPT),
\]

for every \(i = 0, 1, \ldots, k-1\). Moreover, by the definition of function \(\Phi_i\), we get \(\Phi_k(S_k) = g(S_k) - \ell(S_k)\) and \(\Phi_0(S_0) = (1 - 1/k)^k \cdot g(\emptyset) \geq 0\). It follows that

\[
E_{S_k}[g(S_k) - \ell(S_k)] = E_{S_0}[\Phi_0(S_0)] + \sum_{i=0}^{k-1} \left[ E_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - E_{S_i}[\Phi_i(S_i)] \right] \\
\geq \sum_{i=0}^{k-1} \left[ \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-1} g(OPT) - \frac{1}{k} \ell(OPT) \right] \\
= (1 - 1/k)^{k-1} \cdot g(OPT) - \ell(OPT) \geq 1/e \cdot g(OPT) \\
- \ell(OPT),
\]

where the last inequality holds due to \(\ln(1 - 1/k) \geq -1/(k - 1)\) for every \(k \geq 2\).

As for the number of value oracle queries, we notice that \(O(n)\) value oracle queries are needed at each iteration. Thus, during \(k\) iterations, Algorithm 2 requires \(O(kn)\) value oracle queries in total. \(\square\)

### 4.2. Distorted random sampling greedy

Note that at each iteration Algorithm 2 finds an element whose distorted marginal contribution is among top \(r (r \leq k)\) maximum contributions, which will require \(O(n)\) value oracle queries. In order to reduce the number of oracle queries, Mirzasoleiman et al. \[27\] and Buchbinder et al. \[28\] suggested that instead of searching the entire ground set \(\Omega\), we may at each iteration look for a proper element in a relatively small-sized subset of \(\Omega\), which is obtained via random sampling. We adopt this idea, and propose our second algorithm (i.e. Algorithm 3) for the regularized nonmonotone submodular maximization under a cardinality constraint.

Algorithm 3 uses a parameter \(\epsilon\) to control the tradeoff between the number of value oracle queries and the approximation ratio of the solution it returns. Formally, such tradeoff can be seen in the following theorem.

**Theorem 4.5:** There exists a randomized algorithm that given a nonmonotone submodular function \(g: 2^\Omega \rightarrow \mathbb{R}_+\), a normalized modular function \(\ell: 2^\Omega \rightarrow \mathbb{R}_+\), and parameters \(k \geq 2\) and \(\epsilon \in (0, 1/e)\), returns a feasible solution \(S \subseteq \Omega\) with

\[
E_S[g(S) - \ell(S)] \geq (1/e - \epsilon) \cdot g(OPT) - \ell(OPT),
\]

where \(OPT \in \arg\max\{g(A) - \ell(A) : A \subseteq \Omega, |A| \leq k\}\). Moreover, the algorithm performs \(O((n/e^2) \ln(1/\epsilon))\) value oracle queries.
Next, we explain the reason why Algorithm 3 requires parameter $\epsilon$ to be located in the interval $(\delta_k, 1/e)$. Obviously, when $\epsilon \geq 1/e$, Theorem 4.5 provides no approximation guarantee due to $1/e - \epsilon \leq 0$. On the other hand, when $\epsilon \in (0, \delta_k]$, $k \leq (8/e^2) \ln(2/e)$ holds due to the monotonicity of $(8/x^2) \ln(2/x)$, which suggests that the number of value oracle queries Algorithm 2 requires is $O(kn) = O((8n/e^2) \ln(2/e)) = O((n/e^2) \ln(1/e))$. Thus, in the case of $\epsilon \in (0, \delta_k]$, Algorithm 2 meets the requirements of Theorem 4.5. We only need to consider the case of $\epsilon \in (\delta_k, 1/e)$ in what follows. Now, we begin to analyse the performance guarantee of Algorithm 2. Observe that when $\epsilon \in (\delta_k, 1/e)$, we have $p \in (0, 1]$ and $1 \leq s \leq \lceil pn \rceil$. Suppose $S_i$ is given, we sort all the elements of $\Omega = \{v_1, v_2, \ldots, v_n\}$ in order of nonincreasing distorted marginal gain, i.e. $(1 - 1/k)^{k-i+1} \cdot g(v_p|S_i) - \ell_{v_p} \geq (1 - 1/k)^{k-i+1} \cdot g(v_q|S_i) - \ell_{v_q}$ for every $1 \leq p < q \leq n$. Moreover, we define $n$ random variables $X_j (j \in [n])$ as

$$X_j := \begin{cases} 1, & \text{if } e_i = v_j; \\ 0, & \text{otherwise.} \end{cases}$$

Then, we can get the following two lemmas for the same reason as is shown in the proof of Lemma 4.3 and 4.4 in [28].

**Algorithm 3: Distorted Random Sampling Greedy**

```
Input: function $g$ and $\ell$, cardinality $k$, error $\epsilon \in (\delta_k, 1/e)$ // $\delta_k$ is the unique real number such that $(8/\delta_k^2) \ln(2/\delta_k) = k.$
Output: approximate solution $S_k$
1 $S_0 \leftarrow \emptyset$, $p \leftarrow \frac{8}{ke^2} \ln \frac{2}{e}$, $s \leftarrow \frac{k}{n} \lceil pn \rceil$;
2 for $i = 0$ to $k - 1$ do
3     Let $M_i$ be a uniformly random set containing $\lceil pn \rceil$ elements of $\Omega$;
4     Let $d_i$ be a uniformly random number in the interval $(0, s]$;
5     Let $e_i$ be the element of $M_i$ with the $\lceil d_i \rceil$-th largest distorted marginal contribution to $S_i$;
     // The distorted marginal gain with respect to $S_i$ is equal to $(1 - 1/k)^{k-i+1} \cdot g(e_i|S_i) - \ell_{e_i}$ for every $e \in \Omega$.
6     if $(1 - 1/k)^{k-i+1} \cdot g(e_i|S_i) - \ell_{e_i} > 0$ then
7         $S_{i+1} \leftarrow S_i + e_i$;
8     else
9         $S_{i+1} \leftarrow S_i$;
10    return $S_k$.
```
Lemma 4.6 (Lemma 4.3 in [28]): For every $i = 0, 1, \ldots, k - 1$, it holds that $\mathbb{E}_{e_i}[\sum_{j=1}^{k} X_j|S_i] \geq 1 - \epsilon$.

Lemma 4.7 (Lemma 4.4 in [28]): For every $i = 0, 1, \ldots, k - 1$, it holds that $\mathbb{E}_{e_i}[X_j|S_i]$ is a nonincreasing function of $j$.

We consider the increase of $\Phi$ at each iteration.

Lemma 4.8: At each iteration ($i = 0, 1, \ldots, k - 1$) of Algorithm 3, it holds that

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \Psi_i(S_i, e_i) + \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(S_i).$$

Proof: By the definition of $\Phi$, it holds that

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \left(1 - \frac{1}{k}\right)^{k-(i+1)} \left[ g(S_{i+1}) - g(S_i) \right] + \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} \times g(S_i) - \left[ \ell(S_{i+1}) - \ell(S_i) \right].$$

Consider two cases.

Case 1: $(1 - 1/k)^{k-(i+1)} \cdot g(e_i|S_i) - \ell_{e_i} > 0$. Then, $S_{i+1} = S_i + e_i$ and $\Psi_i(S_i, e_i) = (1 - 1/k)^{k-(i+1)} \cdot g(e_i|S_i) - \ell_{e_i}$. Since $\ell$ is nonnegative, we have $e \notin S_i$, which implies

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(e_i|S_i) - \ell_{e_i}$$

$$+ \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(S_i)$$

$$= \Psi_i(S_i, e_i) + \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(S_i).$$

Case 2: $(1 - 1/k)^{k-(i+1)} \cdot g(e_i|S_i) - \ell_{e_i} \leq 0$. Then, $S_{i+1} = S_i$ and $\Psi_i(S_i, e_i) = 0$.

It follows that

$$\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(S_i)$$

$$= \Psi_i(S_i, e_i) + \frac{1}{k} \left(1 - \frac{1}{k}\right)^{k-(i+1)} g(S_i).$$

The proof is completed. ■
We then lower bound the increase of $\Phi$ at each iteration.

**Lemma 4.9:** For every $i = 0, 1, \ldots, k - 1$, it holds that

$$
\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i}[\Phi_i(S_i)] \\
\geq \frac{1 - \epsilon}{k} \left\{ \left( 1 - \frac{1}{k} \right)^{k-(i+1)} \mathbb{E}_{S_i}[g(OPT \cup S_i)] - \ell(OPT) \right\},
$$

where $OPT \in \text{argmax}\{g(A) - \ell(A) : A \subseteq \Omega, |A| \leq k\}$.

**Proof:** By Lemma 4.8, it holds that

$$
\mathbb{E}_{e_i} [\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i] = \mathbb{E}_{e_i} [\Psi_i(S_i, e_i)|S_i] + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i).
$$

(13)

Consider term $\mathbb{E}_{e_i} [\Psi_i(S_i, e_i)|S_i]$. By the definition of $X_j$, we have $\Psi_i(S_i, e_i) = \sum_{j=1}^n X_j \cdot \Psi_i(S_i, v_j)$. Then, the nonnegativity of $\Psi$ implies $\Psi_i(S_i, e_i) \geq \sum_{j=1}^k X_j \cdot \Psi_i(S_i, v_j)$, which indicates $\mathbb{E}_{e_i} [\Psi_i(S_i, e_i)|S_i] \geq \sum_{j=1}^k \mathbb{E}_{e_i} [X_j|S_i] \cdot \Psi_i(S_i, v_j)$. Since both $\mathbb{E}_{e_i} [X_j|S_i]$ and $\Psi_i(S_i, v_j)$ are nonincreasing functions of $j$, we have

$$
\mathbb{E}_{e_i} [\Psi_i(S_i, e_i)|S_i] \geq \sum_{j=1}^k \mathbb{E}_{e_i} [X_j|S_i] \cdot \frac{1}{k} \sum_{j=1}^k \Psi_i(S_i, v_j) \geq \frac{1 - \epsilon}{k} \sum_{e \in \text{OPT}} \Psi_i(S_i, e)
$$

$$
= \frac{1 - \epsilon}{k} \sum_{e \in \text{OPT}} \max\{0, (1 - 1/k)^{k-(i+1)} \cdot g(e|S_i) - \ell_e\}
$$

$$
\geq \frac{1 - \epsilon}{k} \sum_{e \in \text{OPT}} \left[ (1 - 1/k)^{k-(i+1)} \cdot g(OPT|S_i) - \ell_e \right]
$$

$$
\geq \frac{1 - \epsilon}{k} \left[ (1 - 1/k)^{k-(i+1)} \cdot g(OPT|S_i) - \ell(OPT) \right],
$$

(14)

where the first inequality follows from Chebyshev’s inequality, the second inequality follows from Lemma 4.6, and the last inequality follows from the submodularity of $g$.

Plugging (14) into (13) yields

$$
\mathbb{E}_{e_i} [\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i] \\
\geq \frac{1 - \epsilon}{k} \left\{ \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(OPT|S_i) - \ell(OPT) \right\} + \frac{1}{k} \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(S_i)
$$

$$
\times g(OPT) \\
\geq \frac{1 - \epsilon}{k} \left\{ \left( 1 - \frac{1}{k} \right)^{k-(i+1)} g(OPT \cup S_i) - \ell(OPT) \right\},
$$

(15)
where the last inequality follows from the nonnegativity of $g$. It follows that
\[
\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i} [\Phi_i(S_i)] = \mathbb{E}_{S_i} \left[ \mathbb{E}_{e_i} [\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i)|S_i] \right]
\geq \frac{1 - \epsilon}{k} \left\{ \left( 1 - \frac{1}{k} \right)^{k-1} \mathbb{E}_{S_i}[g(OPT \cup S_i)] - \ell(OPT) \right\},
\]
which concludes the proof.

Next, we lower bound $\mathbb{E}_{S_i}[g(OPT \cup S_i)]$ in terms of $g(OPT)$.

**Lemma 4.10:** For every $i = 0, 1, \ldots, k - 1$, it holds that $\mathbb{E}_{S_i}[g(OPT \cup S_i)] \geq (1 - 1/k)^i \cdot g(OPT)$.

**Proof:** Suppose that $e$ is an arbitrary element in $\Omega$. When event $e \notin S_i$ happens, then $e_i \neq e$ implies $e \notin S_{i+1}$, which indicates
\[
\Pr[e \notin S_{i+1}|e \notin S_i] \geq \Pr[e_i \neq e|e \notin S_i]
= \Pr[e \notin M_i] \cdot \Pr[e_i \neq e|e \notin S_i, e \notin M_i]
+ \Pr[e \in M_i] \cdot \Pr[e_i \neq e|e \notin S_i, e \in M_i].
\]
According to the settings of Algorithm 3, it holds that
\[
\Pr[e \in M_i] = \left( \frac{n - 1}{[pn] - 1} \right) \left( \frac{n}{[pn]} \right) = \frac{[pn]}{n},
\]
and
\[
\Pr[e_i \neq e|e \notin S_i, e \notin M_i] = 1.
\]
Consider the conditional probability $\Pr[e_i \neq e|e \notin S_i, e \in M_i]$. According to Algorithm 3, we have $\Pr[e_i = e|e \notin S_i, e \in M_i] \leq 1/s$, which implies $\Pr[e_i \neq e|e \notin S_i, e \in M_i] \geq 1 - 1/s$. Then, for every $i = 0, 1, \ldots, k - 2$, it holds that
\[
\Pr[e \notin S_{i+1}|e \notin S_i] \geq \left( 1 - \frac{[pn]}{n} \right) + \frac{[pn]}{n} \cdot \left( 1 - \frac{1}{s} \right) = 1 - \frac{[pn]}{sn} = 1 - \frac{1}{k},
\]
where the last equality holds due to $s = \frac{k}{n}[pn]$. It follows that, for every $i \in [k - 1],$
\[
\Pr[e \notin S_i] = \Pr[e \notin S_0, e \notin S_1, \ldots, e \notin S_{i-1}, e \notin S_i]
= \Pr[e \notin S_0] \cdot \Pr[e \notin S_1|e \notin S_0] \cdots \Pr[e \notin S_i|e \notin S_{i-1}]
= \Pr[e \notin S_0] \cdot \Pr[e \notin S_1|e \notin S_0] \cdots \Pr[e \notin S_i|e \notin S_{i-1}] \geq (1 - 1/k)^i,
\]
where the last equality holds since $e \notin S_j$ implies that $e \notin S_r$ for every $r = 0, 1, \ldots, j - 1$. Thus, we have $\Pr[e \in S_i] = 1 - \Pr[e \notin S_i] \leq 1 - (1 - 1/k)^i$ for every $e \in \Omega$ and $i \in [k - 1]$.
Define \( h: 2^\Omega \rightarrow \mathbb{R}_+ \) as \( h(A) := g(OPT \cup A) \) for every \( A \subseteq \Omega \). By Lemma 4.2, we have \( \mathbb{E}_S[h(S_i)] \geq (1 - 1/k)^i \cdot h(\varnothing) \), which indicates \( \mathbb{E}_S[g(OPT \cup S_i)] \geq (1 - 1/k)^i \cdot f(OPT) \) for every \( i \in [k - 1] \). Since \( \mathbb{E}_{S_0}[g(OPT \cup S_0)] = g(OPT) \), the proof is completed. 

The performance guarantee of Algorithm 3 is illustrated in the following lemma.

**Theorem 4.11:** When Algorithm 3 terminates, it returns a feasible set \( S_k \) with
\[
\mathbb{E}_{S_k}[g(S_k) - \ell(S_k)] \geq (1/e - \epsilon) \cdot g(OPT) - \ell(OPT),
\]
where \( OPT \in \arg\max\{g(A) - \ell(A) : A \subseteq \Omega, |A| \leq k\} \). Algorithm 3 requires \( O(n^2 \epsilon^2 \ln 1/\epsilon) \) value oracle queries in total.

**Proof:** According to Lemmas 4.9 and 4.10, we have
\[
\mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i}[\Phi_i(S_i)] \geq \frac{1 - \epsilon}{k} \left[ \left( 1 - \frac{1}{k} \right)^{k-1} g(OPT) - \ell(OPT) \right],
\]
for every \( i = 0, 1, \ldots, k - 1 \). Since \( \Phi_k(S_k) = g(S_k) - \ell(S_k) \) and \( \Phi_0(S_0) = (1 - 1/k)^k \cdot g(\varnothing) \geq 0 \), it holds that
\[
\mathbb{E}_{S_k}[g(S_k) - \ell(S_k)] = \mathbb{E}_{S_0}[\Phi_0(S_0)] + \sum_{i=0}^{k-1} \left( \mathbb{E}_{S_{i+1}}[\Phi_{i+1}(S_{i+1})] - \mathbb{E}_{S_i}[\Phi_i(S_i)] \right)
\geq \frac{1 - \epsilon}{k} \sum_{i=0}^{k-1} \left[ \left( 1 - \frac{1}{k} \right)^{k-1} g(OPT) - \ell(OPT) \right]
= (1 - \epsilon) \left[ \left( 1 - \frac{1}{k} \right)^{k-1} g(OPT) - \ell(OPT) \right]
\geq (1/e - \epsilon) \cdot g(OPT) - \ell(OPT)
\]
where the last inequality holds due to \( \ln(1 - 1/k) \geq -1/(k - 1) \) for every \( k \geq 2 \).

As for the number of value oracle queries, we notice that \( O([pn]) \) value oracle queries are needed at each iteration. Thus, during \( k \) iterations, Algorithm 3 requires \( O(k[pn]) = O((n/\epsilon^2) \ln(1/\epsilon)) \) value oracle queries. 

**5. Unconstrained problem**

In this section, we study the unconstrained problem, i.e.
\[
\max\{g(A) - \ell(A) : A \subseteq \Omega\}.
\]
Since the unconstrained problem is a special case of the cardinality-constrained problem (i.e. \( k = n \)), we may directly apply Algorithm 2 or Algorithm 3 to Problem (15) by setting \( k = n \), which will result in an algorithm whose complexity.
is $O(n^2)$ or $O((n/e^2) \ln(1/e))$. Fortunately, we can utilize the property of Problem (15) and design a much simpler and faster algorithm (i.e. Algorithm 4), which uses only $O(n)$ value oracle queries but enjoys the same approximation guarantee of Algorithm 2 and Algorithm 3. Similar to the analysis of Algorithms 2 and 3, we also need two auxiliary functions $\Phi$ and $\Psi$. For every $i = 0, 1, \ldots, n$ and $T \subseteq \Omega$, define

$$
\Phi_i(T) := (1 - 1/n)^{n-i} \cdot g(T) - \ell(T).
$$

**Algorithm 4: Distorted Random Greedy for Problem (15)**

**Input:** function $g$ and $\ell$

**Output:** approximate solution $S_n$

1. $S_0 \leftarrow \emptyset$;
2. for $i = 0$ to $n - 1$ do
3.   Let $e_i$ be a uniformly random element of $\Omega$;
4.   if $(1 - 1/n)^{n-i} \cdot g(e_i|S_i) - \ell(e_i) > 0$ then
5.     $S_{i+1} \leftarrow S_i + e_i$;
6.   else
7.     $S_{i+1} \leftarrow S_i$;
8. return $S_n$.

In addition, for every $i = 0, 1, \ldots, n - 1$, $T \subseteq \Omega$ and $e \in \Omega$, define

$$
\Psi_i(T, e) := \max\{0, (1 - 1/n)^{n-i} \cdot g(e|T) - \ell(e)\}.
$$

First, we consider the increase of $\Phi$ at each iteration. Similar to the proof of Lemma 4.8, we can obtain the following lemma.

**Lemma 5.1:** At each iteration $(i = 0, 1, \ldots, n - 1)$ of Algorithm 4, it holds that

$$
\Phi_{i+1}(S_{i+1}) - \Phi_i(S_i) = \Psi_i(S_i, e_i) + \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-i} g(S_i).
$$

Next, we consider the lower bound of term $\Psi_i(S_i, e_i)$.

**Lemma 5.2:** At each iteration $(i = 0, 1, \ldots, n - 1)$ of Algorithm 4, it holds that

$$
\mathbb{E}_{(S_i, e_i)}[\Psi_i(S_i, e_i)] \geq \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} g(OPT) - \frac{1}{n} \ell(OPT) - \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-i} \mathbb{E}_{S_i}[g(S_i)],
$$

where $OPT \in \text{argmax}\{g(A) - \ell(A) : A \subseteq \Omega\}$.
Proof: Note that

\[ \mathbb{E}_{e_i}[\Psi_i(S_i, e_i) | S_i] = \sum_{e \in \Omega} \Pr[e_i = e | S_i] \cdot \Psi_i(S_i, e) \]

\[ = \frac{1}{n} \sum_{e \in \Omega} \Psi_i(S_i, e) \geq \frac{1}{n} \sum_{e \in \text{OPT}} \Psi_i(S_i, e) \]

\[ \geq \frac{1}{n} \sum_{e \in \text{OPT}} \left[ \left( 1 - \frac{1}{n} \right)^{n-(i+1)} g(e | S_i) - \ell_e \right] \]

\[ \geq \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-(i+1)} g(\text{OPT} | S_i) - \frac{1}{n} \ell(\text{OPT}), \]

where the first inequality follows from the nonnegativity of \( \Psi_i \), and the last inequality follows from the submodularity of \( g \). It follows that

\[ \mathbb{E}_{(S_i, e_i)}[\Psi_i(S_i, e_i)] = \mathbb{E}_{S_i}[\mathbb{E}_{e_i}[\Psi_i(S_i, e_i) | S_i]] \]

\[ \geq \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-(i+1)} \mathbb{E}_{S_i}[g(\text{OPT} | S_i)] - \frac{1}{n} \ell(\text{OPT}). \]

Following similar argument in the proof of Lemma 4.3, we have \( \mathbb{E}_{S_i}[g(\text{OPT} \cup S_i)] \geq (1 - 1/n)^i \cdot g(\text{OPT}) \) for every \( i = 0, 1, \ldots, n-1 \), which implies

\[ \mathbb{E}_{(S_i, e_i)}[\Psi_i(S_i, e_i)] \geq \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-1} g(\text{OPT}) - \frac{1}{n} \ell(\text{OPT}) \]

\[ - \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-(i+1)} \mathbb{E}_{S_i}[g(S_i)]. \]

The proof is done.

In the same spirit that Lemmas 4.1 and 4.3 imply Theorem 4.4, we can obtain the following theorem, which illustrate the performance guarantee of Algorithm 4, based on Lemmas 5.1 and 5.2.

Theorem 5.3: When Algorithm 4 terminates, it returns a feasible set \( S_n \) with

\[ \mathbb{E}_{S_n}[g(S_n) - \ell(S_n)] \geq 1/e \cdot g(\text{OPT}) - \ell(\text{OPT}), \]

where \( \text{OPT} \in \arg\max \{ g(A) - \ell(A) : A \subseteq \Omega \} \). Algorithm 4 performs \( O(n) \) value oracle queries in total.

6. Conclusion

In this paper, we study the regularized nonmonotone submodular maximization problem thoroughly. We propose several algorithms for the optimization
problem subject to various constraints, including matroid constraint, cardinality constraint and no constraint. We give a systematic and unified analysis of the approximation guarantee and time complexity of the algorithms we propose. According to the analysis, those algorithms are both effective and efficient.

Note
1. See Section 2 for more details.
2. Namely, the algorithm returns a solution $S \in \mathcal{F}$ such that $g(S) - \ell(S) \geq \alpha \cdot \max\{g(A) - \ell(A) : A \in \mathcal{F}\}$, where $\alpha$ is a constant in $(0, 1]$.
3. In Tables 1 and 2, $S^*$ represents an optimal solution to the corresponding optimization problem.
4. See line 5 of Algorithm 1.
5. Throughout this paper, $[k]$ denotes set $\{1, 2, \ldots, k\}$.
6. Every set $A$ obeying the cardinality constraint is an independent set of a uniform matroid.
7. If $a_1 \geq a_2 \geq \cdots \geq a_n$ and $b_1 \geq b_2 \geq \cdots \geq b_n$, then it holds that $\frac{1}{n} \sum_{k=1}^{n} a_k b_k \geq \left( \frac{1}{n} \sum_{k=1}^{n} a_k \right) \cdot \left( \frac{1}{n} \sum_{k=1}^{n} b_k \right)$.

Acknowledgments
We are indebted to the associate editor and an anonymous reviewer for their valuable comments and suggestions, which help us improve the quality of this paper.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Funding
This research was supported by the National Natural Science Foundation of China [grant numbers 11991022 and 12071459] and the Fundamental Research Funds for the Central Universities [grant number E1E40107X2].

References
[1] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2003. p. 137–146.
[2] Krause A, Guestrin C. Near-optimal observation selection using submodular functions. In: AAAI; Vol. 7; 2007. p. 1650–1654.
[3] Lin H, Bilmes J. 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; 2010. p. 912–920.
[4] Lehmann B, Lehmann D, Nisan N. Combinatorial auctions with decreasing marginal utilities. Games Econ Behav. 2006;55(2):270–296.
[5] Vondrák J. Optimal approximation for the submodular welfare problem in the value oracle model. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing; 2008. p. 67–74.
[6] Grötschel M, Lovász L, Schrijver A. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica. 1981;1(2):169–197.

[7] Schrijver A. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J Comb Theory Ser B. 2000;80(2):346–355.

[8] Iwata S, Fleischer L, Fujishige S. A combinatorial strongly polynomial algorithm for minimizing submodular functions. J ACM. 2001;48(4):761–777.

[9] Feige U. A threshold of ln n for approximating set cover. J ACM. 1998;45(4):634–652.

[10] Krause A, Guestrin C. Near-optimal nonmyopic value of information in graphical models; 2012. arXiv preprint arXiv:12071394.

[11] Nikolakaki S, Ene A, Terzi E. An efficient framework for balancing submodularity and cost. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining; 2021. p. 1256–1266.

[12] Kazemi E, Minaee S, Feldman M, et al. Regularized submodular maximization at scale. In: International Conference on Machine Learning. PMLR; 2021. p. 5356–5366.

[13] Nemhauser GL, Wolsey LA, Fisher ML. An analysis of approximations for maximizing submodular set functions-I. Math Program. 1978;14(1):265–294.

[14] Fisher ML, Nemhauser GL, Wolsey LA. An analysis of approximations for maximizing submodular set functions-II. In: Polyhedral combinatorics. Springer; 1978. p. 73–87.

[15] Nemhauser GL, Wolsey LA. Best algorithms for approximating the maximum of a submodular set function. Math Oper Res. 1978;3(3):177–188.

[16] Calinescu G, Chekuri C, Pal M, et al. Maximizing a monotone submodular function subject to a matroid constraint. SIAM J Comput. 2011;40(6):1740–1766.

[17] Buchbinder N, Feldman M, Naor J, et al. Submodular maximization with cardinality constraints. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM; 2014. p. 1433–1452.

[18] Gharan SO, Vondrák J. Submodular maximization by simulated annealing. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM; 2011. p. 1098–1116.

[19] Feldman M, Naor J, Schwartz R. A unified continuous greedy algorithm for submodular maximization. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. IEEE; 2011. p. 570–579.

[20] Buchbinder N, Feldman M, Seffi J, et al. A tight linear time (1/2)-approximation for unconstrained submodular maximization. SIAM J Comput. 2015;44(5):1384–1402.

[21] Feige U, Mirrokni VS, Vondrák J. Maximizing non-monotone submodular functions. SIAM J Comput. 2011;40(4):1133–1153.

[22] Sviridenko M, Vondrák J, Ward J. Optimal approximation for submodular and supermodular optimization with bounded curvature. Math Oper Res. 2017;42(4):1197–1218.

[23] Feldman M. Guess free maximization of submodular and linear sums. Algorithmica. 2021;83(3):853–878.

[24] Harshaw C, Feldman M, Ward J, et al. Submodular maximization beyond non-negativity: guarantees, fast algorithms, and applications. In: International Conference on Machine Learning. PMLR; 2019. p. 2634–2643.

[25] Vondrák J. Symmetry and approximability of submodular maximization problems. SIAM J Comput. 2013;42(1):265–304.

[26] Alon N, Spencer JH. The probabilistic method. Hoboken, New Jersey: John Wiley & Sons; 2004.

[27] Mirzasoleiman B, Badanidiyuru A, Karbasi A, et al. Lazier than lazy greedy. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 29; 2015.

[28] Buchbinder N, Feldman M, Schwartz R. Comparing apples and oranges: query trade-off in submodular maximization. Math Oper Res. 2017;42(2):308–329.