Journal of Combinatorial Optimization (2023) 45:96
https://doi.org/10.1007/s10878-023-01026-5

Algorithms for maximizing monotone submodular function minus modular function under noise

Shufang Gong[1] · Bin Liu[1] · Mengxue Geng[1] · Qizhi Fang[1]

Accepted: 24 March 2023 / Published online: 19 April 2023

Abstract
Submodular function has the property of diminishing marginal gain, and thus it has a wide range of applications in combinatorial optimization and in emerging disciplines such as machine learning and artificial intelligence. For any set S, most of previous works usually do not consider how to compute $f(S)$, but assume that there exists an oracle that will output $f(S)$ directly. In reality, however, the process of computing the exact f is often inevitably inaccurate or costly. At this point, we adopt the easily available noise version F of f. In this paper, we investigate the problems of maximizing a non-negative monotone normalized submodular function minus a non-negative modular function under the ε-multiplicative noise in three situations, i.e., the cardinality constraint, the matroid constraint and the online unconstraint. For the above problems, we design three deterministic bicriteria approximation algorithms using greedy and threshold ideas and furthermore obtain good approximation guarantees.

Keywords Submodular minus modular · Multiplicative noise · Bicriteria algorithm · Cardinality constraint · Matroid constraint

1 Introduction

The submodular function has the property of diminishing marginal gain, that is, the marginal gain of any element added to a given set cannot exceed that of its subset. The position of submodular optimization in combinatorial optimization corresponds to that of convex optimization in continuous optimization. A set function $f : 2^G \to \mathbb{R}$ is *submodular* iff $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for any subsets $A, B \subseteq G$.

This work was supported in part by the National Natural Science Foundation of China (11971447, 11871442), and the Fundamental Research Funds for the Central Universities.

✉ Bin Liu
binliu@ouc.edu.cn

[1] School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China

Another equivalent definition is that for any subsets $S \subseteq T \subseteq G$ and any element $e \in G \setminus T$, $f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$. A set function f is *monotone* if for any $S \subseteq T$, it holds $f(S) \leq f(T)$; and f is *normalized* if $f(\emptyset) = 0$. For convenience, $f_A(B) \triangleq f(A \cup B) - f(A)$ and $F_A(B) \triangleq F(A \cup B) - F(A)$. Similarly, $f_A(j) \triangleq f(A \cup \{j\}) - f(A)$ and $F_A(j) \triangleq F(A \cup \{j\}) - F(A)$.The goal of the submodular optimization problem is to find a feasible solution among a finite number of solutions to achieve the optimum or near-optimum.

Submodular minimization with unconstraint can be solved in polynomial time (Grötschel et al. 2012; Iwata et al. 2001). Submodular minimization with constraints is very difficult, even under very simple constraints (Iwata and Nagano 2009; Koufogiannakis and Young 2013; Svitkina and Fleischer 2011). In this paper, we focus on submodular maximization problems. Submodular maximization problems are usually NP-hard, so many works have studied approximation algorithms for the corresponding problems. In addition, submodular maximization has a wide range of application scenarios, such as sensor placement (Krause et al. 2006), clustering (Liu et al. 2013), profit maximization with multiple adoptions (Zhang et al. 2016), subset selection (Das and Kempe 2018), secretary problems (Bateni et al. 2013), and so on.

In many practical problems, it is desirable to seek a feasible solution that can balance benefits and costs, such as team formation problem, influence maximization problem and recommender systems problem etc. To be specific, take the influence maximization problem as an example. A company wants to select a set of representatives from the community to promote its products and to sell as many products as possible while employing as little cost as possible. In other words, find a subset to maximize the company's net profit. These problems can be modeled as $max_{S \in \mathscr{I}} f(S) - c(S)$ where $\mathcal{I}$ is some constraint. Under the noise-free environment, there has been a great deal of work (Harshaw et al. 2019; Nikolakaki et al. 2021; Qian 2021; Sviridenko et al. 2017; Wang et al. 2021). Although f and c are both non-negative, $f - c$ can be potentially negative. In Feige (1998) and Papadimitriou and Yannakakis (1991) it is shown that there is no multiplicative approximation guarantee in polynomial time for the problems of possibly negative submodular maximization with or without constraints. Similar to previous works (Du et al. 2014; Wang et al. 2021), we use a weaker approximation: S is (α, β)-*bicriteria approximate* if

$$f(S) - c(S) \geq \alpha \cdot f(S^*) - \beta \cdot c(S^*),$$

where $\alpha \in [0, 1]$, $\beta \geq 0$ and S^* is an optimal solution.

In some real-world problems, it is costly to compute the exact value of the submodular function or a slight error inevitably arises in the process, such as revealed preference theory (Chambers and Echenique 2016), crowdsourced image collection summarization (Singla et al. 2016), active learning (Feldman 2009), information maximization (Chen et al. 2015), etc. Thus we take the noise version F to approximately replace f for calculation, where F is easy to calculate. This paper focuses on the ε-*multiplicative noise version* F of f, i.e., for any set S, it holds

$$(1 - \varepsilon) f(S) \leq F(S) \leq (1 + \varepsilon) f(S).$$

Although f is monotone submodular, F is *not* necessarily monotone or submodular. But we assume that F is normalized in this paper. Therefore, some methods used to solve the submodular maximization problem in the noise-free environment cannot be directly applied to the noisy version, which brings challenges to our work. Fortunately, we can use the ease of computation of F to design algorithms and the approximately submodularity of F to analyze the quality of $f(S)$.

Our contribution For the problem of maximizing a non-negative monotone normalized submodular function minus a non-negative modular function with noise in three situations: the cardinality constraint, the matroid constraint and the online unconstraint, we devise three concise deterministic algorithms and obtain good approximation guarantees.

2 Related work

First, we list the relevant theoretical results that the objective function is a submodular function f minus a modular function c. Note that $f - c$ is still submodular.

The cardinality constraint For maximizing a non-negative monotone γ-weakly submodular function minus a non-negative modular function, Harshaw et al. (2019) proposed a deterministic algorithm with approximation ratio $(1 - e^{-\gamma}, 1)$ and a stochastic algorithm with approximation ratio $(1 - e^{-\gamma} - O(\varepsilon), 1)$ using greedy idea. For the same problem, Qian (2021) designed a stochastic algorithm with approximation ratio $(1 - e^{-\gamma}, 1)$ using the multi-objective evolutionary method. For maximizing a non-negative monotone submodular function minus a non-negative modular function, Nikolakaki et al. (2021) devised a more concise deterministic algorithm with approximation ratio $(\frac{1}{2}, 1)$ and used the lazy evaluations technique to further accelerate its algorithm.

The matroid constraint For maximizing a non-negative monotone submodular function minus a non-negative modular function, Sviridenko et al. (2017) designed a stochastic algorithm with approximation ratio $(1 - e^{-\gamma}, 1)$. Due to the high computational cost of Sviridenko et al. (2017) and Nikolakaki et al. (2021) proposed a deterministic algorithm with approximation ratio $(\frac{1}{2}, 1)$ and better query complexity.

No constraints For maximizing a non-negative monotone submodular function minus a non-negative modular function, Sviridenko et al. (2017) designed a stochastic algorithm with approximation ratio $(1 - e^{-\gamma}, 1)$. Later, Nikolakaki et al. (2021) proposed a deterministic algorithm with approximation ratio $(\frac{1}{2}, 1)$.

Next, we describe some related works under noisy models. Submodular maximization with noise is closely related to approximately submodular maximization. F is called ε-approximately submodular if there is a submodular function f such that $(1 - \varepsilon) f(S) \leq F(S) \leq f(S)$ for any $S \subseteq G$. The difference is that the objective of the submodular maximization problem with noise is f while the objective of the approximately submodular maximization problem is F. Horel and Singer (2016) first introduced the ε-approximately submodular maximization. For maximizing a monotone submodular function, Horel and Singer (2016) designed an algorithm to achieve the $1 - e^{-1} - O(\delta)$ approximation ratio using greedy ideas, where $\varepsilon \leq \frac{1}{k}$, $\delta = \varepsilon k$. Later, Gölz and Procaccia (2019) considered the problem of submodular maximization with ε-multiplicative noise under the P-matroids constraint and devised a $(P + 1 + \frac{4k\varepsilon}{1-\varepsilon})^{-1}$-approximation algorithm where k is the size of maximum feasible set of the P-matroids. Yang et al. (2019) studied the problem of submodular maximization under the streaming model with two types of noise: multiplicative noise and additive noise. When noise tends to 0, the approximation ratio both can reach $\frac{2}{k}$ where k is the cardinality. For the same problem, Xiao et al. (2021) considered it with the differential privacy noise and got an approximation ratio close to $\frac{1}{(2+(1+\frac{1}{k})^2)(1+\frac{1}{k})}$.

3 The bicriteria algorithm with the cardinality constraint under noise

In this section, we consider the first problem

$$\max_{S \subseteq G} \; f(S) - c(S)$$
$$\text{s.t.} \quad |S| \leq k$$

where f is a non-negative monotone normalized submodular and c is a non-negative modular function. In Algorithm 1, we use the distorted objective function about the noise function F. For convenience, we define $\Phi_i(S) = (1 - \frac{1}{k})^{k-i} F(S) - xc(S)$ for any subset $S \subseteq G$ and any iteration $i = 1, 2, \ldots, k$. We define $\Psi_i(S, e) = \max\{0, (1 - \frac{1}{k})^{k-i} F_S(e) - xc(e)\}$ for any subset $S \subseteq G$, any element $e \in G$ and any iteration $i = 1, 2, \ldots, k$. At each iteration, Algorithm 1 selects an element e_i with the maximum marginal gain of the distorted objective function, and then this algorithm only accepts e_i if it has positive marginal gain. Algorithm 1 terminates after k iterations. Denote the set after i iterations as S_i and the output set S_k satisfies $|S_k| \leq k$ according to the iteration rules.

Algorithm 1 The distorted bicriteria algorithm for $f - c$ under noise with the cardinality constraint

Input: Ground set G, noisy function F, non-negative modular function c, cardinality constraint $k \in \mathbf{N}_+$, noise parameter ϵ and $x = 1 + \epsilon + 2\epsilon k$
Output: Set S_k

1: Initially set $S_0 := \emptyset$
2: **for** $i = 1, 2, \cdots, k$ **do**
3: $\quad e_i = \arg\max_{e \in G \setminus S_{i-1}} \{(1 - \frac{1}{k})^{k-i} F_{S_{i-1}}(e) - xc(e)\}$
4: $\quad$ **if** $(1 - \frac{1}{k})^{k-i} F_{S_{i-1}}(e) - xc(e) > 0$ **then**
5: $\quad\quad S_i := S_{i-1} \cup \{e_i\}$
6: $\quad$ **else**
7: $\quad\quad S_i := S_{i-1}$
8: $\quad$ **end if**
9: **end for**
10: **return** S_i

In this section, we can analyze the approximation ratio of the above algorithm by the following two lemmas. Lemmas 1 and 2 together indicate that the marginal gain of distorted objective at each iteration has a lower bound to ensure the desired approximate ratio.

Lemma 1 *In each iteration of Algorithm 1, we have*

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

Proof Based on the definition of the function Φ, we have

$$\begin{aligned}
&\Phi_i(S_i) - \Phi_{i-1}(S_{i-1}) \\
=&(1-\frac{1}{k})^{k-i}F(S_i) - xc(S_i) - (1-\frac{1}{k})^{k-(i-1)}F(S_{i-1}) + xc(S_{i-1}) \\
=&(1-\frac{1}{k})^{k-i}F(S_i) - (1-\frac{1}{k})(1-\frac{1}{k})^{k-i}F(S_{i-1}) - xc(S_i) + xc(S_{i-1}) \\
=&(1-\frac{1}{k})^{k-i}\big(F(S_i) - F(S_{i-1})\big) + \frac{1}{k}(1-\frac{1}{k})^{k-i}F(S_{i-1}) - x\big(c(S_i) - c(S_{i-1})\big).
\end{aligned}$$

The following discussion is divided into two situations. Firstly, if the elements e_i is added to the set S_{i-1} in the i th iteration, then we have

$$\Psi_i(S_{i-1}, e_i) = (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e_i) - xc(e_i) > 0.$$

And $F(S_i)-F(S_{i-1}) = F(S_{i-1}\cup e_i)-F(S_{i-1}) = F_{S_{i-1}}(e_i), c(S_i)-c(S_{i-1}) = c(e_i)$. Therefore, we can get

$$\begin{aligned}
\Phi_i(S_i) - \Phi_{i-1}(S_{i-1}) &= (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e_i) - xc(e_i) + \frac{1}{k}(1-\frac{1}{k})^{k-i}F(S_{i-1}) \\
&= \Psi_i(S_{i-1}, e_i) + \frac{1}{k}(1-\frac{1}{k})^{k-i}F(S_{i-1}).
\end{aligned}$$

Secondly, if the elements e_i is not added to the set S_{i-1}, then $\Psi_i(S_{i-1}, e_i) = 0 \geq (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e_i) - xc(e_i)$ and $S_i = S_{i-1}$. Obviously, $F(S_i) - F(S_{i-1}) = 0$, $c(S_i) - c(S_{i-1}) = 0$. Then we obtain

$$\begin{aligned}
\Phi_i(S_i) - \Phi_{i-1}(S_{i-1}) &= 0 + \frac{1}{k}(1-\frac{1}{k})^{k-i}F(S_{i-1}) \\
&= \Psi_i(S_{i-1}, e_i) + \frac{1}{k}(1-\frac{1}{k})^{k-i}F(S_{i-1}).
\end{aligned}$$

In summary, at each iteration, it holds that

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

□

Next, we discuss the lower bound of the function $\Psi_i(S_{i-1}, e_i)$ through Lemma 2.

Lemma 2 *S_k is the output solution of Algorithm 1, we have*

$$\Psi_i(S_{i-1}, e_i) \geq \frac{(1-\epsilon)}{k}(1-\frac{1}{k})^{k-i}[f(S^*)-f(S_{i-1})]-2\epsilon(1-\frac{1}{k})^{k-i}f(S_k)-\frac{x}{k}c(S^*).$$

Proof Denote by S^* an optimal solution of the first problem. By the definition of Ψ,

$$\begin{aligned}
k\cdot\Psi_i(S_{i-1}, e_i) &= k\cdot\max\{0, (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e_i) - xc(e_i)\} \\
&= k\cdot\max\{0, \max_{e\in G\setminus S_{i-1}}[(1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)]\} \\
&= k\cdot\max\{0, \max_{e\in G}[(1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)]\} \\
&= k\cdot\max_{e\in G}\{0, (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)\} \\
&\geq |S^*|\cdot\max_{e\in G}\{0, (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)\} \\
&\geq |S^*|\cdot\max_{e\in S^*}\{0, (1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)\} \\
&\geq \sum_{e\in S^*}[(1-\frac{1}{k})^{k-i}F_{S_{i-1}}(e) - xc(e)] \\
&= (1-\frac{1}{k})^{k-i}\sum_{e\in S^*}[F(S_{i-1}\cup\{e\}) - F(S_{i-1})] - xc(S^*) \\
&\geq (1-\frac{1}{k})^{k-i}\sum_{e\in S^*}[(1-\epsilon)f(S_{i-1}\cup\{e\})-(1+\epsilon)f(S_{i-1})]-xc(S^*) \\
&= (1-\epsilon)(1-\frac{1}{k})^{k-i}\sum_{e\in S^*}[f(S_{i-1}\cup\{e\}) - f(S_{i-1})] \\
&\quad - 2\epsilon(1-\frac{1}{k})^{k-i}\sum_{e\in S^*}f(S_{i-1}) - xc(S^*).
\end{aligned} \tag{1}$$

By the submodularity and monotonicity of f, we have

$$\sum_{e\in S^*}[f(S_{i-1}\cup\{e\}) - f(S_{i-1})] \geq f(S_{i-1}\cup S^*) - f(S_{i-1}) \geq f(S^*) - f(S_{i-1}).$$

Clearly, $S_{i-1}\subseteq S_k$ and by the monotonicity of f, we have $f(S_{i-1}) \leq f(S_k)$. Therefore, inequality (1) becomes

$$\begin{aligned}
k\cdot\Psi_i(S_{i-1}, e_i) \geq\ & (1-\epsilon)(1-\frac{1}{k})^{k-i}[f(S^*) - f(S_{i-1})] \\
& - 2\epsilon k(1-\frac{1}{k})^{k-i}f(S_k) - xc(S^*).
\end{aligned}$$

Both sides of the above inequality are divided by k at the same time,then

$$\Psi_i(S_{i-1}, e_i) \geq \frac{(1-\epsilon)}{k}(1-\frac{1}{k})^{k-i}[f(S^*) - f(S_{i-1})] - 2\epsilon(1-\frac{1}{k})^{k-i} f(S_k) - \frac{x}{k}c(S^*).$$

Thus, we complete the proof. □

Combined with Lemmas 1 and 2, we can get the Theorem 1 as follows.

Theorem 1 *When $x = 1 + \epsilon + 2\epsilon k$, Algorithm 1 returns the set S_k such that*

$$f(S_k) - c(S_k) \geq \frac{1-\epsilon}{1+\epsilon+2\epsilon k}(1-e^{-1}) f(S^*) - c(S^*).$$

If the parameter $\epsilon \to 0$, the approximate ratio is $(1-e^{-1}, 1)$.

Proof On the one hand,by Lemmas 1 and 2, we have

$$\begin{aligned}
&\Phi_i(S_i) - \Phi_{i-1}(S_{i-1}) \\
\geq &\frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i}[f(S^*) - f(S_{i-1})] - 2\epsilon(1-\frac{1}{k})^{k-i} f(S_k) \\
&- \frac{x}{k}c(S^*) + \frac{1}{k}(1-\frac{1}{k})^{k-i} F(S_{i-1}) \\
\geq &\frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i} f(S^*) - \frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i} f(S_{i-1}) - 2\epsilon(1-\frac{1}{k})^{k-i} f(S_k) \\
&- \frac{x}{k}c(S^*) + \frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i} f(S_{i-1}) \\
= &\frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i} f(S^*) - 2\epsilon(1-\frac{1}{k})^{k-i} f(S_k) - \frac{x}{k}c(S^*).
\end{aligned}$$

On the other hand, by the definition of the function Φ, it is clear that

$$\Phi_0(S_0) = (1-\frac{1}{k})^k F(\emptyset) - xc(\emptyset) \geq 0$$

and

$$\Phi_k(S_k) = (1-\frac{1}{k})^0 F(S_k) - xc(S_k).$$

We can get

$$\begin{aligned}
F(S_k) - xc(S_k) &\geq \Phi_k(S_k) - \Phi_0(S_0) \\
&= \sum_{i=1}^{k} \big(\Phi_i(S_i) - \Phi_{i-1}(S_{i-1})\big) \\
&\geq \sum_{i=1}^{k} \Big(\frac{1-\epsilon}{k}(1-\frac{1}{k})^{k-i} f(S^*) - 2\epsilon(1-\frac{1}{k})^{k-i} f(S_k) - \frac{x}{k}c(S^*)\Big) \\
&= \frac{1-\epsilon}{k}\sum_{i=1}^{k}(1-\frac{1}{k})^{k-i} f(S^*) - 2\epsilon\sum_{i=1}^{k}(1-\frac{1}{k})^{k-i} f(S_k) - \frac{x}{k}\sum_{i=1}^{k} c(S^*).
\end{aligned}$$

Since

$$\sum_{i=1}^{k}(1-\frac{1}{k})^{k-i} = k\big(1-(1-\frac{1}{k})^k\big),$$

we have

$$\begin{aligned}
F(S_k) - xc(S_k) &= \frac{1-\epsilon}{k}k\big(1-(1-\frac{1}{k})^k\big) f(S^*) \\
&\quad - 2\epsilon k\big(1-(1-\frac{1}{k})^k\big) f(S_k) - xc(S^*) \\
&\geq (1-\epsilon)\big(1-(1-\frac{1}{k})^k\big) f(S^*) - 2\epsilon k f(S_k) - xc(S^*).
\end{aligned}$$

Based on the definition of function F, we can get

$$\begin{aligned}
(1+\epsilon) f(S_k) - xc(S_k) &\geq F(S_k) - xc(S_k) \\
&\geq (1-\epsilon)\big(1-(1-\frac{1}{k})^k\big) f(S^*) - 2\epsilon k f(S_k) - xc(S^*)
\end{aligned}$$

Arranging the above inequality, then

$$(1+\epsilon+2\epsilon k) f(S_k) - xc(S_k) \geq (1-\epsilon)\big(1-(1-\frac{1}{k})^k\big) f(S^*) - xc(S^*).$$

Thus, we have

$$\begin{aligned}
f(S_k) - c(S_k) &\geq \frac{1-\epsilon}{1+\epsilon+2\epsilon k}\big(1-(1-\frac{1}{k})^k\big) f(S^*) - c(S^*) \\
&\geq \frac{1-\epsilon}{1+\epsilon+2\epsilon k}(1-e^{-1}) f(S^*) - c(S^*).
\end{aligned}$$

Completing the proof. □

4 The bicriteria algorithm with the matroid constraint under noise

In this section, we consider the second problem

$$\begin{aligned} &\max_{S \subseteq G} \; f(S) - c(S) \\ &\text{s.t.} \quad S \in \mathscr{F} \end{aligned}$$

where f is a non-negative monotone normalized submodular, c is a non-negative modular function and the constraint is a matroid constraint. Before analyzing the specific approximation guarantee of Algorithm 2, we firstly introduce the definition of matroid $\mathscr{M} = (G, \mathscr{F})$ by Edmonds (2003).

Definition 1 (Edmonds 2003)
Given a finite ground set G and a collection of subsets $\mathscr{F} \subseteq 2^G$, $(G, \mathscr{F})$ is called a matroid iff this pair satisfies:

(1) $\emptyset \in \mathscr{F}$;
(2) Hereditary property: if for any subsets I, J satisfying $I \subseteq J$ and $J \in \mathscr{F}$, then $I \in \mathscr{F}$;
(3) Augmentation property: if for any subsets I, J satisfying $I, J \in \mathscr{F}$ and $|J| > |I|$, then there exists an element $e \in J \setminus I$ such that $I \cup \{e\} \in \mathscr{F}$.

In the following, we list a common result presented by Brualdi (1969) on matroid.

Algorithm 2 The bicriteria algorithm for $f - c$ under noise with the matroid constraint

Input: Ground set G, noisy function F, non-negative modular function c, matroid $(G, \mathscr{F})$, noise parameter ϵ, $x = 2 + 2\epsilon + 4\epsilon r$ and r is the rank of the matriod $(G, \mathscr{F})$.
Output: Set S'

Initially set $S_0 := \emptyset$, $M := G$
for $i = 1, 2, \cdots, n$ **do**
  if $M = \emptyset$ **then**
    break
  else
    $e_i = \arg\max_{e \in M}\{F(e|S_{i-1}) - xc(e)\}$
    if $F(e_i|S_{i-1}) - xc(e_i) > 0$ **then**
      $S_i := S_{i-1} \cup \{e_i\}$
    else
      break
    end if
  end if
  delete all the elements in M such that $S_{i-1} \cup \{e\} \notin \mathscr{F}$
end for
return S' where $S' = S_j$ for some $j \in \{1, 2, \ldots, n\}$

Lemma 3 (Brualdi 1969)
For any $A \in \mathscr{F}$ and any $B \in \mathscr{F}$, if they satisfy $|A| = |B|$, then there exists a bijection $\phi : A \backslash B \to B \backslash A$ such that $(B \backslash \phi(e)) \cup \{e\} \in \mathscr{F}$ for any $e \in A \backslash B$.

Denote S' and M' as the output solution set and the set of elements left from G at the end of Algorithm 2, respectively. Denote the current solution set after i iterations as S_i and the element added in i-th iteration as e_i. Suppose that S^* is an optimal solution of the second problem and the rank of matroid $(G, \mathscr{F})$ is r. To facilitate the analysis of the approximation ratio, we use surrogate function $h(S) = F(S) - xc(S)$ replace the original objective function $f(S) - c(S)$ in our algorithm where x can be determined later.

We divide the optimal solution S^* into two parts S_1^* and S_2^* where $S_1^* = S^* \cap M'$ and $S_2^* = S^* \setminus S_1^*$. The following lemma tells us that the number of elements in S' is not less than the number of elements in S_2^*. It guarantees the existence of S_i for $i = 1, 2, \ldots, |S_2^*|$.

Lemma 4 *We have*

$$|S'| \geq |S_2^*|.$$

Proof Since S' and S^* are bases of the matroid and $S_2^* \subseteq S^*$, then $|S'| \geq |S_2^*|$ is obvious. □

Let $l = |S_2^*|$. By Lemma 4, we know that S_l exists. Then, we analyze the relationship between S' and S_2^* and prove the following conclusion.

Lemma 5 *It holds that*

$$(1 - \epsilon) f(S_l \cup S_2^*) - 2\epsilon r f(S') - xc(S_2^*) \leq 2 f(S_l) - xc(S_l). \tag{2}$$

Proof Since $|S_l| = |S_2^*| = l$, $S_2^* \in \mathscr{F}$ and $S_l \in \mathscr{F}$, by Lemma 3, we can establish a bijection $\tau : S_2^* \rightarrow S_l$ such that $S_l \setminus \tau(e) \cup \{e\} \in \mathscr{F}$ for each element $e \in S_2^* \setminus S_l$ and $\tau(e) = e$ for each element $e \in S_2^* \cap S_l$.

For any $i = 1, 2, \cdots, l$, denote by $S_2^{*(i)} = \tau^{-1}(S_i)$ and $s_i^* = \tau^{-1}(e_i)$. By the selection rule, we get that

$$h_{S_{i-1}}(e_i) \geq h_{S_{i-1}}(s_i^*), \tag{3}$$

which is showed as follows. In the above bijection, if $s_i^* = e_i$, the inequality obviously established. If $s_i^* \neq e_i$, i.e. $s_i^* \in S_2^* \setminus S_l$, we know $S_l \backslash \{e_i\} \cup \{s_i^*\} \in \mathscr{F}$. For each $j < i$, we have $S_j \subseteq S_l \backslash \{e_i\}$, thus $S_j \cup \{s_i^*\} \in \mathscr{F}$ by the hereditary property. Thus element s_i^* is a candidate for e_i, and the inequality $h_{S_{i-1}}(e_i) \geq h_{S_{i-1}}(s_i^*)$ holds.

Thus, for each $i = 1, \cdots, l$, we expand the inequality (3) by definition to obtain

$$F(S_i) - F(S_{i-1}) - xc(e_i) \geq F(S_{i-1} \cup \{s_i^*\}) - F(S_{i-1}) - xc(s_i^*). \tag{4}$$

Based on the definition of F and the submodularity of function f, the right-hand side of the inequality (4) can be rewritten as

$$\begin{aligned}
&F(S_{i-1}\cup\{s_i^*\})-F(S_{i-1})-xc(s_i^*)\\
&\geq(1-\epsilon)f(S_{i-1}\cup\{s_i^*\})-(1+\epsilon)f(S_{i-1})-xc(s_i^*)\\
&=(1-\epsilon)\Big(f(S_{i-1}\cup\{s_i^*\})-f(S_{i-1})\Big)-2\epsilon f(S_{i-1})-xc(s_i^*)\\
&\geq(1-\epsilon)\Big(f(S_l\cup s_i^*)-f(S_l)\Big)-2\epsilon f(S_{i-1})-xc(s_i^*)\\
&\geq(1-\epsilon)\Big(f(S_l\cup S_2^{*(i)})-f(S_l\cup S_2^{*(i-1)})\Big)-2\epsilon f(S_{i-1})-xc(s_i^*).
\end{aligned}$$

Combing (4) and a fact that f is monotone, we have

$$\begin{aligned}
F(S_i)-F(S_{i-1})-xc(e_i)\geq(1-\epsilon)&\Big(f(S_l\cup S_2^{*(i)})-f(S_l\cup S_2^{*(i-1)})\Big)\\
&-2\epsilon f(S')-xc(s_i^*).
\end{aligned}$$

Summing up all $i=1,\cdots,l$,

$$\begin{aligned}
(1+\epsilon)f(S_l)-xc(S_l)&\geq F(S_l)-F(\emptyset)-xc(S_l)\\
&\geq(1-\epsilon)\Big(f(S_l\cup S_2^*)-f(S_l)\Big)-2\epsilon\sum_{i=1}^{l}f(S')-xc(S_2^*)\\
&\geq(1-\epsilon)\Big(f(S_l\cup S_2^*)-f(S_l)\Big)-2\epsilon rf(S')-xc(S_2^*).
\end{aligned}$$

Rearranging the above inequality, we can get

$$(1-\epsilon)f(S_l\cup S_2^*)-2\epsilon rf(S')-xc(S_2^*)\leq 2f(S_l)-xc(S_l).$$

□

On the other hand, there is also a relationship between S' and S_1^*.

Lemma 6 *It holds that*

$$(1-\epsilon)f(S'\cup S_1^*)-xc(S_1^*)\leq(1-\epsilon+2\epsilon r)f(S'). \tag{5}$$

Proof Without loss of generality, we assume $S_1^*\neq\emptyset$. Since $S_1^*=\emptyset$ obviously holds. According to the selection rule in Algorithm 2, for each $s^*\in S_1^*$, it satisfies

$$h_{S'}(s^*)=h(S'\cup\{s^*\})-h(S')\leq 0.$$

That is

$$F(S'\cup\{s^*\})-F(S')-xc(s^*)\leq 0.$$

Thus, we have

$$\sum_{s^* \in S_1^*} \Big(F(S' \cup \{s^*\}) - F(S') \Big) \leq \sum_{s^* \in S_1^*} xc(s^*). \tag{6}$$

By the definition of F, we get

$$(1-\epsilon) \sum_{s^* \in S_1^*} \Big(f(S' \cup \{s^*\}) - f(S') \Big) - 2\epsilon \sum_{s^* \in S_1^*} f(S') \leq \sum_{s^* \in S_1^*} xc(s^*).$$

By the submodularity of f, we have

$$(1-\epsilon) f_{S'}(S_1^*) - 2\epsilon r f(S') \leq xc(S_1^*),$$

Rearranging it we can get Lemma 6. □

By the above two Lemmas, we get Theorem 2 as follows.

Theorem 2 *When $x = 2 + 2\epsilon + 4\epsilon r$, Algorithm 2 returns a solution $S' \in \mathscr{F}$ such that*

$$f(S') - c(S') \geq \frac{1-\epsilon}{2 + 2\epsilon + 4\epsilon r} f(S^*) - c(S^*),$$

where r is the rank of matroid $(G, \mathscr{F})$. If the parameter $\epsilon \to 0$, the approximate ratio is $(\frac{1}{2}, 1)$.

Proof Based on the definition of submodular function and $S^* = S_1^* \cup S_2^*$, $S_l \subseteq S'$, we obtain

$$f(S_l \cup S_1^*) + f(S' \cup S_2^*) \geq f(S_l) + f(S' \cup S^*). \tag{7}$$

Summing (2) and (5). After utilizing (7), we get

$$(1-\epsilon) f(S^*) - xc(S^*) \leq (1 - \epsilon + 4\epsilon r) f(S') + (1+\epsilon) f(S_l) - xc(S_l). \tag{8}$$

According to the rule of Algorithm 2, we know that for any element $e_i \in S'$, it obeys the condition that

$$h(S_i) - h(S_{i-1}) > 0,$$

then

$$\sum_{i=l+1}^{|S'|} \Big(h(S_i) - h(S_{i-1}) \Big) = h(S') - h(S_l) > 0.$$

By the definition of function h and F, we have

$$(1+\epsilon) f(S') - xc(S') \geq (1-\epsilon) f(S_l) - xc(S_l).$$

Combing the inequality (8), it holds that

$$
\begin{aligned}
(1-\epsilon)f(S^*) - xc(S^*) &\le (1-\epsilon+4\epsilon r)f(S') + (1+\epsilon)f(S_l) - xc(S_l)\\
&= (1-\epsilon+4\epsilon r)f(S') + (1-\epsilon)f(S_l) + 2\epsilon f(S_l) - xc(S_l)\\
&\le (1-\epsilon+4\epsilon r)f(S') + (1+\epsilon)f(S') - xc(S') + 2\epsilon f(S_l).
\end{aligned}
$$

Rearranging to get the following inequality

$$(2+4\epsilon r)f(S') + 2\epsilon f(S_l) - xc(S') \ge (1-\epsilon)f(S^*) - xc(S^*).$$

By $S_l \subseteq S'$ and the monotonicity of f, we get

$$(2+2\epsilon+4\epsilon r)f(S') - xc(S') \ge (1-\epsilon)f(S^*) - xc(S^*).$$

When $x = 2+2\epsilon+4\epsilon r$, the above inequality is obviously satisfied. Hence, we get the final approximation guarantee:

$$f(S') - c(S') \ge \frac{1-\epsilon}{2+2\epsilon+4\epsilon r} f(S^*) - c(S^*).$$

□

5 The bicriteria algorithm with online unconstrained problem under noise

In this section, we consider the last problem

$$\max_{S\subseteq G} f(S) - c(S)$$

where f is a non-negative monotone normalized submodular, c is a non-negative modular function and G is a online set. In the online setting, the elements arrive one at a time in the form of stream. At this time, we need to decide whether to add it to the current solution set, and this decision is irrevocable. Here, we consider this model in a noisy environment which means that using the easily calculable noise function F.

Denote by S^* an optimal solution of the last problem and S' is the output solution of Algorithm 3. Let $|S^*| = q$ and $\tilde{S} = S^* \setminus S' = \{s_1, \cdots, s_{|\tilde{S}|}\}$. In this algorithm, we use the surrogate function $h(S) = F(S) - xc(S)$ and the value of x can be determined later. For each iteration, we add the arriving element to the current solution only when its marginal gain exceeds 0. Furthermore, let the first i elements of $\tilde{S}$ be $\tilde{S}_i$, i.e., $\tilde{S}_i = \{s_1, \cdots, s_i\} \subseteq \tilde{S}$ for all $1 \le i \le |\tilde{S}|$. The details of the bicriteria algorithm are as follows.

We analyze the approximation ratio by the following lemma.

Algorithm 3 The online bicriteria algorithm for $f - c$ under noise

Input: Ground set $G = \{e_1, e_2, \cdots, e_n\}$, noisy function F, non-negative modular function c, noise parameter ϵ and $x = 2\Big(1 + \frac{2\epsilon n}{1+\epsilon}\Big)$

Output: Set S'

Initially set $S_0 := \emptyset$
for each arriving element e_i **do**
  if $F_{S_{i-1}}(e_i) - xc(e_i) > 0$ **then**
    $S_i := S_{i-1} \cup \{e_i\}$
  else
    $S_i := S_{i-1}$
  end if
end for
return S' where $S' = S_n$

Lemma 7 *It holds that*

$$h(S' \cup \tilde{S}) - h(S') \leq 2\epsilon n f(S^*) + \frac{4\epsilon n}{1-\epsilon} f(S') + \frac{2\epsilon}{1-\epsilon} xc(S^*) \tag{9}$$

Proof It is clearly that

$$h(S' \cup S^*) - h(S') = h(S' \cup \tilde{S}) - h(S') = \sum_{i=1}^{|\tilde{S}|} \Big(h(S' \cup \tilde{S}_i) - h(S' \cup \tilde{S}_{i-1}) \Big).$$

Then,

$$\begin{aligned}
& h(S' \cup \tilde{S}_i) - h(S' \cup \tilde{S}_{i-1}) \\
&= F(S' \cup \tilde{S}_i) - xc(S' \cup \tilde{S}_i) - F(S' \cup \tilde{S}_{i-1}) + xc(S' \cup \tilde{S}_{i-1}) \\
&\leq (1+\varepsilon) f(S' \cup \tilde{S}_i) - xc(S' \cup \tilde{S}_i) - (1-\varepsilon) f(S' \cup \tilde{S}_{i-1}) + xc(S' \cup \tilde{S}_{i-1}) \\
&\leq (1+\epsilon)\Big(f(S' \cup \tilde{S}_i) - f(S' \cup \tilde{S}_{i-1}) \Big) + 2\epsilon f(S' \cup S^*) - xc(s_i) \\
&\leq (1+\epsilon) f_{S'}(s_i) + 2\epsilon f(S') + 2\epsilon f(S^*) - xc(s_i) \\
&\leq (1+\epsilon)\Big(\frac{F(S' \cup \{s_i\})}{1-\epsilon} - \frac{F(S')}{1+\epsilon} \Big) + 2\epsilon f(S') + 2\epsilon f(S^*) - xc(s_i) \\
&\leq \frac{1+\epsilon}{1-\epsilon} F(S' \cup \{s_i\}) - F(S') + 2\epsilon f(S') + 2\epsilon f(S^*) - xc(s_i) \\
&\leq \frac{1+\epsilon}{1-\epsilon}\Big(h(S' \cup \{s_i\}) - h(S') \Big) + \frac{4\epsilon}{1-\epsilon} f(S') + 2\epsilon f(S^*) + \frac{2\epsilon}{1-\epsilon} xc(s_i).
\end{aligned} \tag{10}$$

where the second inequality used $\tilde{S}_{i-1} \subseteq \tilde{S} \subseteq S^*$ and the monotonicity of f. The third inequality obeys the submodularity of function f. Thus, there is

$$\begin{aligned} & h(S' \cup \tilde{S}) - h(S') \\ & = \sum_{i=1}^{|\tilde{S}|} \Big(h(S' \cup \tilde{S}_i) - h(S' \cup \tilde{S}_{i-1}) \Big) \\ & \le \frac{1+\epsilon}{1-\epsilon} \sum_{i=1}^{|\tilde{S}|} \Big(h(S' \cup \{s_i\}) - h(S') \Big) + \frac{4\epsilon n}{1-\epsilon} f(S') + 2\epsilon n f(S^*) + \frac{2\epsilon}{1-\epsilon} x c(S^*) \end{aligned}$$

According to the selection rule of the Algorithm 3, the marginal contribution from elements that are not added to S' is less than 0, i.e.

$$\sum_{i=1}^{|\tilde{S}|} \Big(h(S' \cup \{s_i\}) - h(S') \Big) \le 0.$$

Then, we complete the proof of lemma 7. □

Based on lemma 7, we analyse the approximation ratio as follows.

Theorem 3 *When $x = 2\Big(1 + \frac{2\epsilon n}{1+\epsilon}\Big)$, the Algorithm 3 returns a solution S' such that*

$$f(S') - c(S') \ge \frac{(1-\epsilon-2\epsilon n)(1-\epsilon)}{2+2\epsilon+4\epsilon n} f(S^*) - c(S^*), \tag{11}$$

If the parameter $\epsilon \to 0$, we have $(\frac{1}{2}, 1)$-approximation ratio.

Proof On the one hand, lemma 7 illustrates the upper bound of $h(S' \cup \tilde{S}) - h(S')$. On the other hand, we give the lower bound

$$\begin{aligned} h(S' \cup \tilde{S}) - h(S') & \ge (1-\epsilon) f(S' \cup \tilde{S}) - (1+\epsilon) f(S') - x\Big(c(S' \cup \tilde{S}) - c(S') \Big) \\ & \ge (1-\epsilon) f(S^*) - (1+\epsilon) f(S') - x c(S^*), \end{aligned}$$

where the first inequality used the definition of F, the second inequality used the monotonicity of function f and c. Combing the lemma 7, it is obvious that

$$\frac{1-\epsilon^2+4\epsilon n}{1-\epsilon} f(S') \ge (1-\epsilon-2\epsilon n) f(S^*) - \frac{1+\epsilon}{1-\epsilon} x c(S^*),$$

i.e.

$$f(S') \ge \frac{(1-\epsilon-2\epsilon n)(1-\epsilon)}{1-\epsilon^2+4\epsilon n} f(S^*) - \frac{1+\epsilon}{1-\epsilon^2+4\epsilon n} x c(S^*).$$

Finally, we consider the value of x. Since the algorithm only adds elements whose marginal contribution exceeds 0, then $h(S') - h(\emptyset) > 0$, i.e., $h(S') > 0$. Thus

$$c(S') < \frac{1+\epsilon}{x} f(S') \quad (x > 0).$$

Rearranging, we can get

$$\begin{aligned} f(S') - c(S') &> f(S') - \frac{1+\epsilon}{x} f(S') \\ &\geq \left(1 - \frac{1+\epsilon}{x}\right)\left(\frac{(1-\epsilon-2\epsilon n)(1-\epsilon)}{1-\epsilon^2+4\epsilon n} f(S^*) - \frac{1+\epsilon}{1-\epsilon^2+4\epsilon n} x c(S^*)\right). \end{aligned} \tag{12}$$

When $x = 2\left(1 + \frac{2\epsilon n}{1+\epsilon}\right)$, then

$$f(S') - c(S') \geq \frac{(1-\epsilon-2\epsilon n)(1-\epsilon)}{2+2\epsilon+4\epsilon n} f(S^*) - c(S^*).$$

It completes the proof of this Theorem. □

6 Conclusions

In this paper, we study the problem of monotone submodular functions minus modular functions with ϵ-multiplicative noise under different constraints. For the first problem, i.e., the monotone submodular minus modular function under the cardinality constraint, Algorithm 1 constructs the distorted surrogate function and gives the $(\frac{1-\epsilon}{1+\epsilon+2\epsilon k}(1-e^{-1}), 1)$ approximation ratio using the greedy idea. For the second problem, i.e., the monotone submodular minus modular function under the matroid constraint, Algorithm 2 constructs a concise surrogate function and gives the $(\frac{1-\epsilon}{2+2\epsilon+4\epsilon r}, 1)$ approximation ratio using the greedy idea. For the last problem, i.e., the monotone submodular minus modular function with online unconstraint, Algorithm 3 constructs a similarly concise surrogate function to give the $(\frac{(1-\epsilon-2\epsilon n)(1-\epsilon)}{2+2\epsilon+4\epsilon n}, 1)$ approximation ratio by adding only the elements whose marginal contributions exceed 0. When $\epsilon \to 0^+$, thus we get $(1-e^{-1}, 1)$-approximation ratio, $(\frac{1}{2}, 1)$-approximation ratio and $(\frac{1}{2}, 1)$-approximation ratio to these problems respectively.

Funding The Funding was provided by National Natural Science Foundation of China (grant no: 11971447, 11871442) and Fundamental Research Funds for the Central Universities.

Data availability Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare that they have no competing interest.

References

Bateni MH, Hajiaghayi M, Zadimoghaddam M (2013) Submodular secretary problem and extensions. ACM Trans Algorithms 9(4):1–23

Brualdi RA (1969) Comments on bases in dependence structures. Bull Aust Math Soc 1(2):161–167

Chambers CP, Echenique F (2016) Revealed preference theory. Cambridge University Press, England

Chen Y, Hassani SH, Karbasi A (2015) Sequential information maximization: when is greedy near-optimal? In: Proceedings of The 28th Conference on Learning Theory, pp. 338-363. Paris, France

Das A, Kempe D (2018) Approximate submodularity and its applications: subset selection, sparse approximation and dictionary selection. J Mach Learn Res 19(1):74–107

Du DL, Li Y, Xiu NH (2014) Simultaneous approximation of multi-criteria submodular function maximization. J Oper Res Soc China 2(3):271–290

Edmonds J (2003) Submodular functions, matroids, and certain polyhedra. In: Jünger M, Reinelt G, Rinaldi G (eds) Combinatorial optimization ? Eureka, You Shrink!. Lecture notes in computer science, vol. 2570, pp. 11–26. Springer, Berlin, Heidelberg

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

Feldman V (2009) On the power of membership queries in agnostic learning. J Mach Learn Res 10:163–182

Gölz P, Procaccia AD (2019) Migration as submodular optimization. In: Proceedings of the 33rd AAAI conference on artificial intelligence, pp. 549-556. Honolulu, Hawaii, USA

Grötschel M, Lovász L, Schrijver A (2012) Geometric algorithms and combinatorial optimization. Springer-Verlag, Berlin, Germany

Harshaw C, Feldman M, Ward J (2019) Submodular maximization beyond non-negativity: guarantees, fast algorithms, and applications. In: Proceedings of the 36th international conference on machine learning, pp. 2634–2643. Long Beach, California, USA

Horel T, Singer Y (2016) Maximization of approximately submodular functions. In: Proceedings of the 30th international conference on neural information processing systems, pp. 3045-3053. Barcelona, Spain

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

Iwata S, Nagano K (2009) Submodular function minimization under covering constraints. In: Proceedings of the 50th annual IEEE symposium on foundations of computer science, pp. 671–680. Atlanta, Georgia, USA

Koufogiannakis C, Young NE (2013) Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular cost. Algorithmica 66(1):113–152

Krause A, Leskovec J, Isovitsch S (2006) Optimizing sensor placements in water distribution systems using submodular function maximization. In: Proceedings of the 8th international conference on water distribution systems analysis symposium. pp. 1–17. Cincinnati, Ohio

Liu MY, Tuzel O, Ramalingam S (2013) Entropy-rate clustering: cluster analysis via maximizing a submodular function subject to a matroid constraint. IEEE Trans Pattern Anal Mach Intell 36(1):99–112

Nikolakaki SM, Ene A, Terzi E (2021) An efficient framework for balancing submodularity and cost. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining, pp. 1256–1266. Virtual Event/Singapore

Papadimitriou CH, Yannakakis M (1991) Optimization, approximation, and complexity classes. J Comput Syst Sci 43(3):425–440

Qian C (2021) Multiobjective evolutionary algorithms are still good: maximizing monotone approximately submodular minus modular functions. Evol Comput 29(4):463–490

Singla A, Tschiatschek S, Krause A (2016) Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In: Proceedings of the 30th AAAI conference on artificial intelligence. pp. 12-17. Phoenix, Arizona, USA

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

Svitkina Z, Fleischer L (2011) Submodular approximation: sampling-based algorithms and lower bounds. SIAM J Comput 40(6):1715–1737

Wang Y, Xu Y, Yang X (2021) On maximizing the difference between an approximately submodular function and a linear function subject to a matroid constraint. In: Proceedings of the 15th international conference on combinatorial optimization and applications, pp. 75–85. Tianjin, China

Xiao D, Guo L, Liao K (2021) Streaming submodular maximization under differential privacy noise. In: Proceedings of the 15th international conference on combinatorial optimization and applications. pp. 431–444. Tianjin, China

Yang R, Xu D, Cheng Y (2019) Streaming submodular maximization under noises. In: Proceedings of the 39th international conference on distributed computing systems, pp. 348–357

Zhang H, Zhang H, Kuhnle A (2016) Profit maximization for multiple products in online social networks. In: Proceedings of the 35th annual IEEE international conference on computer communications, pp. 1–9