Two-Stage Non-submodular Maximization

Hong Chang[1], Zhicheng Liu[2], Ping Li[3], and Xiaoyan Zhang[1](✉)

[1] School of Mathematical Science and Institute of Mathematics,
Nanjing Normal University, Nanjing 210023, China
{changh,zhangxiaoyan}@njnu.edu.cn, royxyzhang@gmail.com
[2] Beijing Institute for Scientific and Engineering Computing,
Beijing University of Technology, Beijing 100124, China
[3] Shandong Qiguang Information Technology Co. Ltd., Jiading Qu 253000, China
liping@vista.aero

Abstract. The concept of submodularity finds wide applications in data science, artificial intelligence, and machine learning, providing a boost to the investigation of new ideas, innovative techniques, and creative algorithms to solve different submodular optimization problems arising from a diversity of applications. However pure submodular problems only represent a small portion of the problems we are facing in real life applications. To solve these optimization problems, an important research method is to describe the characteristics of the non-submodular functions. The non-submodular functions is a hot research topic in the study of nonlinear combinatorial optimizations. In this paper, we combine and generalize the curvature and the generic submodularity ratio to design an approximation algorithm for two-stage non-submodular maximization under a matroid constraint.

Keywords: Two-stage submodular maximization · Matroid constraint · Curvature · Generic submodularity ratio

1 Introduction

Submodular function maximization has drawn much attention practical and theoretical interests [5,6,11,13]. For a given set V, the function $f : 2^V \to \mathbb{R}$ is said to be submodular if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ for $\forall X, Y \subseteq V$. A set function f is called monotone if $f(X) \leq f(Y)$ for all $X \subseteq Y \subseteq V$ and it is said to be normalized when $f(\emptyset) = 0$. The well-known greedy algorithm presents a constraint-factor approximation ratio $1 - 1/e$ for submodular maximization subject to a cardinality constraint [10]. The bounds can be improved if one make further assumptions on submodular functions. For example, the curvature of a submodular function f in [3] is defined as

$$k_f = 1 - \min_{v \in V} \frac{f(V) - f(V \backslash \{v\})}{f(v)},$$

D.-Z. Du et al. (Eds.): TAMC 2022, LNCS 13571, pp. 272–280, 2022.
https://doi.org/10.1007/978-3-031-20350-3_22

and noting that the curvature is computable with a linear number of function oracle calls, then the greedy algorithm obtains $\frac{1}{k_f}(1-e^{-k_f})$ guarantee under a cardinality constraint [3].

The greedy algorithm is a simple and effective technique to solve many optimization problems. However, the ground set is often so large that the well-known greedy algorithm is not enough efficient. One solution to the problem is to give some training functions to reduce the ground set, and then Balkanski et al. [1] gave the concept of the two-stage submodular maximization problem. For a ground set V and a constant k, the objective is to obtain a set $S \subseteq V$ of size at most k and m subsets $T_1, T_2, \ldots, T_m$ in $\mathcal{I}(S)$ to maximize the following

$$\frac{1}{m}\sum_{i=1}^{m}\max_{T\in\mathcal{I}(S)} f_i(T),$$

where $f_i : 2^V \to \mathbb{R}_+$ is submodular for $i = 1, \ldots, m$, and $\mathcal{I}(S)$ is a constraint set over the reduced ground set $S \subseteq V$.

Related works have been conducted in the area of the two-stage submodular maximization. For a cardinality constraint, that is $\mathcal{I}(S)$= $\{T : |T| \leq k\}$. When k is enough large, Balkanski et al. [1] used the continuous optimization method to design an approximation algorithm with approximation ratio, which asymptotically approaches $1 - 1/e$. When k is small, a local search algorithm was obtained with approximation ratio close to $1/2$ in [1]. In addition, Mitrovic et al. [9] considered the two-stage submodular maximization with cardinality constraint under streaming and distributed settings. For a matriod constraint, Stan et al. [12] obtained a new local-search based algorithm with approximation ratio $1 - 1/e^2$.

On the other hand, for many applications in practice, including experimental design and sparse Gaussian processes [8], the objective function is in general not submodular. The results for submodular optimization problems are not no longer maintained. To solve these optimization problems, an important research method is to introduce some parameters to describe the characteristics of the non-submodular functions, such as submodularity ratio, curvature, generic submodularity ratio, and then design algorithms for the problems and analyze the performances of the algorithms with these parameters. Given a ground set V and a nondecreasing set function $f : 2^V \to \mathbb{R}$, the generic submodularity ratio of f is the largest scalar γ such that for any $X \subseteq Y \subseteq V$ and any $v \in V \setminus Y$, $f(v|X) \geq \gamma f(v|Y)$, which is a quantity characterizing how close a nonnegative nondecreasing set function is to be submodular [4]. And, a function is called γ-submodular if its generic submodularity ratio is γ. A natural curvature notion can also be introduced for non-submodular functions. We recall that the curvature of a non-negative set function [2] is the smallest scalar α such that for $\forall\ S,\ T \subseteq V,\ i \in S \setminus T$,

$$f(i|S \setminus \{i\} \cup T) \geq (1-\alpha) f(i|S \setminus \{i\}).$$

Based on the above motivation, we discuss the two-stage γ-submodular maximization problem under a matriod constraint. Our main contribution is to design an approximation algorithm with constant approximation ratio with respect to the curvature and the generic submodularity ratio. The rest of our paper is summarized as below. In Sect. 2, we show some technical preliminaries, including notations and relevant known results. In Sect. 3, we give an approximation algorithms along with its analysis. And some concluding remarks are presented in Sect. 4.

2 Preliminaries

Firstly, we recall the following known concepts and results for submodular functions, supmodular functions and modulars function.

Definition 1. *For a given set V, the function $f : 2^V \to \mathbb{R}$ is called submodular if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ for $\forall X, Y \subseteq V$.*

An equivalent definition is that the function $f : 2^V \to \mathbb{R}$ is said to be submodular if $f(e|S) \geq f(e|T)$ for $S \subseteq T \subset V$ and $v \in V \setminus S$, where $f(e|S) = f(e \cup S) - f(S)$.

Definition 2. *For a given set V, the function $f : 2^V \to \mathbb{R}$ is called modular if $f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$ for $\forall X, Y \subseteq V$.*

Furthermore, we define the concept of modular functions.

Definition 3. *For a given set V, the function $f : 2^V \to \mathbb{R}$ is called modular if $f(X) + f(Y) = f(X \cap Y) + f(X \cup Y)$ for $\forall X, Y \subseteq V$.*

Next, we formally restate the two-stage submodular maximization problem. For a ground set V, and m nonnegative, monotone and normalized γ-submodular functions $f_1, \ldots, f_m$, which are drawn from some unknown distribution, our aim is to select a set $S \subseteq V$ of size at most k and m subsets $T_1, \ldots, T_m$ in $\mathcal{I}(S)$ to maximize the following

$$\frac{1}{m} \sum_{i=1}^{m} \max_{T \in \mathcal{I}(S)} f_i(T), \tag{2.1}$$

where $\mathcal{I}(S)$ is a constraint set over $S \subseteq V$. In our paper, the set $\mathcal{I}(S)$ corresponds to a matriod constraint.

Definition 4. *For a given set S and $\mathcal{I} \in 2^S$, a matroid $\mathcal{M} = (S, \mathcal{I})$ satisfies three properties: (1) $\emptyset \in \mathcal{I}$; (2) if $P \subseteq Q \in \mathcal{I}$, then $P \in \mathcal{I}$; (3) if $P, Q \in \mathcal{I}$ and $|P| \leq |Q|$, then $P + q \in \mathcal{I}$, where $q \in Q \backslash P$.*

The mapping below is a very useful tool to study the matriod constraint, which is shown in [7].

Proposition 1. *Let $\mathcal{M}_i = (S, \mathcal{I}_i)$ be a matroid for $i \in \{1, \ldots, p\}$. For $\forall X, Y \in \mathcal{I}_i$, there is a mapping $\pi_i : Y \backslash X \to X \backslash Y \cup \{\emptyset\}$, which satisfies the three properties (1) $(X \setminus \pi_i(y)) \cup y \in \mathcal{I}_i$ for $\forall y \in Y \setminus X$; (2) $|\pi_i^{-1}(x)| \leq 1$ for $\forall x \in X \setminus Y$; (3) let $X_y = \{\pi_1(y), \ldots, \pi_p(y)\}$, then $(X \setminus X_y) \cup y \in \cap_{i=1}^{p} \mathcal{I}_i$ for $\forall y \in Y \setminus X$.*

Now, we turn to give a nice result for γ-submodular functions in [4].

Proposition 2. *Let f be a γ-submodular function. Then for $\forall S \subseteq T$,*

$$f(T) \leq f(S) + \frac{1}{\gamma} \sum_{v \in T \setminus S} f(v|S).$$

To maximize the above objective function, we discuss the following

$$\ell_i(T) = (1-\alpha) \sum_{v \in T} f_i(v),$$
$$g_i(T) = f_i(T) - \ell_i(T).$$

where the function $\ell_i(T)$ is modular, and the function $g_i(T)$ is a monotone and normalized γ-submodular function, this is because that the function $f_i(T)$ is monotone and normalized.

Problem (2.1) turns into the following problem: obtain a set $S \subseteq V$ of size at most k and m subsets $T_1, T_2, \ldots, T_m$ in $\mathcal{I}(S)$ to maximize the following

$$\frac{1}{m} \sum_{i=1}^{m} \max_{T \in \mathcal{I}(S)} (g_i(T) + \ell_i(T)).$$

For notational convenience, we use the following notations. In terms of the function g_i, define the marginal gain of adding an element x to the set T_i^j as

$$\Delta_i^g(x, T_i^j) = g_i(\{x\} \cup T_i^j) - g_i(T_i^j).$$

Similarly, the gain of replacing y with x in terms of the set T_i^j is represented by

$$\nabla_i^g(x, y, T_i^j) = g_i(\{x\} \cup T_i^j \setminus \{y\}) - g_i(T_i^j).$$

Furthermore, for the functions g_i and ℓ_i, we define

$$\Delta_i(x, T_i^j) = \left(1 - \frac{\gamma + \frac{1}{\gamma}}{k}\right)^{k-j} \Delta_i^g(x, T_i^j) + \left(1 - \frac{1}{k}\right)^{k-j} \ell_i(x),$$

$$\nabla_i(x, y, T_i^j) = \left(1 - \frac{\gamma + \frac{1}{\gamma}}{k}\right)^{k-j} \nabla_i^g(x, y, T_i^j) + \left(1 - \frac{1}{k}\right)^{k-j} (\ell_i(x) - \ell_i(y)).$$

The set of elements in T_i^j can replace x, which will not violate the matroid constraint, is defined as

$$\mathcal{I}(x, T_i^j) = \{y \in T_i^j : T_i^j \cup \{x\} \setminus \{y\} \in \mathcal{I}(S)\}.$$

Based on the notations of $\Delta_i(x, T_i^j)$ and $\nabla_i(x, y, T_i^j)$, we denote the replacement gain of x in terms of T_i^j by

$$\nabla_i(x, T_i^j) = \begin{cases} \Delta_i(x, T_i^j) & \text{if}\, T_i^j \cup \{x\} \in \mathcal{I}(S), \\ \max\{0, \max_{y \in \mathcal{I}(x, T_i^j)} \nabla_i(x, y, T_i^j)\} & \text{otherwise.} \end{cases}$$

In addition, we define

$$\mathsf{Rep}_i(x, T_i^j) = \begin{cases} \emptyset & \text{if} T_i^j \cup \{x\} \in \mathcal{I}(S), \\ \arg\max_{y \in \mathcal{I}(x, T_i^j)} \nabla_i(x, y, T_i^j) & \text{otherwise.} \end{cases}$$

3 Problem (2.1) Under a Matroid Constraint

In this section, we discuss Problem (2.1) under $\mathcal{I}(S)$ is a matroid constraint. A replacement greedy algorithm is shown in Sect. 3.1, and then we analyze its approximation ratio in Sect. 3.2.

3.1 Algorithm

Our replacement greedy algorithm starts with $S^0 = \emptyset$, and runs in k rounds. In each round, a new element can be added into the current solution if it does not violate the matroid constraints or can be replaced with some element in the current set while increasing the value of the objective function.

Algorithm 1.

```
$S^0 \leftarrow \emptyset, T_i^0 \leftarrow \emptyset (\forall 1 \le i \le m)$
for $1 \le j \le k$ do
    $t^* \leftarrow \arg\max_{t \in V} \sum_{i=1}^m \nabla_i(t, T_i^{j-1})$
    $S^j \leftarrow S^{j-1} \cup \{t^*\}$
    for $1 \le i \le m$ do
        if $\nabla_i(t^*, T_i^{j-1}) > 0$ then
            $T_i^j \leftarrow T_i^{j-1} \cup \{t^*\} \setminus \mathsf{Rep}_i(t^*, T_i^{j-1})$
        else
            $T_i^j \leftarrow T_i^{j-1}$
        end if
    end for
end for
Return sets $S^k$ and $T_1^k, T_2^k, \cdots, T_m^k$
```

3.2 Theoretical Analysis

We analyze the performance guarantee of Algorithm 1, which depends on the distorted objective function as follows.

$$\Phi_j(S^j) = \sum_{i=1}^m \left(\left(1 - \frac{\gamma + \frac{1}{\gamma}}{k}\right)^{k-j} g_i(T_i^j) + \left(1 - \frac{1}{k}\right)^{k-j} \ell_i(T_i^j) \right).$$

Lemma 1. *In each iteration of Algorithm 1,*

$$\begin{aligned}&\Phi_j(S^j) - \Phi_{j-1}(S^{j-1})\\ &= \sum_{i=1}^{m}\left(\nabla_i(t^j, T_i^{j-1}) + \frac{\gamma+\frac{1}{\gamma}}{k}\left(1-\frac{\gamma+\frac{1}{\gamma}}{k}\right)^{k-j} g_i(T_i^{j-1}) + \frac{1}{k}\left(1-\frac{1}{k}\right)^{k-j}\ell_i(T_i^{j-1})\right).\end{aligned}$$

For the second term on the right side in Lemma 1, we give the lower bound in the following.

Lemma 2. *If the element $t^j \in V$ is added into the current set S^{j-1}, then*

$$\sum_{i=1}^{m}\nabla_i\left(t^j, T_i^{j-1}\right) \geq \frac{1}{k}\sum_{i=1}^{m}\sum_{t\in T_i^*\setminus T_i^{j-1}}\nabla_i(t, T_i^{j-1}),$$

where $S^* = \underset{S\subseteq V, |S|\leq k}{\arg\max}\sum_{i=1}^{m}\max_{T\in\mathcal{I}(S)} f_i(T)$, $T_i^* = \arg\max_{A\in\mathcal{I}(S^*)} f_i(A)$.

The following lemma is crucial to analyze the approximation ratio of Algorithm 1.

Lemma 3. *For $j = 1, 2, \ldots, k$, we have*

$$\begin{aligned}&\sum_{i=1}^{m}\nabla_i(t^j, T_i^{j-1})\\ &\geq \frac{1}{k}\left(1-\frac{\gamma+\frac{1}{\gamma}}{k}\right)^{k-j}\sum_{i=1}^{m}\left(\gamma(1-\alpha)g_i(T_i^*) - (\gamma+\frac{1}{\gamma})g_i(T_i^{j-1})\right)\\ &\quad+\frac{1}{k}\left(1-\frac{1}{k}\right)^{k-j}\sum_{i=1}^{m}\left(\ell_i(T_i^*) - \ell_i(T_i^{j-1})\right).\end{aligned}$$

Combining Lemma 1 and Lemma 3, the following theorem is proved as below.

Theorem 1. *Algorithm 1 returns a set S^k of size k such that*

$$\sum_{i=1}^{m}\left(g_i(T_i^k) + \ell_i(T_i^k)\right) \geq \frac{\gamma}{\gamma+\frac{1}{\gamma}}\left(1-e^{-(\gamma+\frac{1}{\gamma})}\right)\sum_{i=1}^{m} g_i(T_i^*) + \left(1-e^{-1}\right)\sum_{i=1}^{m}\ell_i(T_i^*).$$

Proof. By the definition of the function Φ, it is obtained that

$$\begin{aligned}&\Phi_0(S^0) = 0,\\ &\Phi_k(S^k) = \sum_{i=1}^{m}\left(\left(1-\frac{\gamma+\frac{1}{\gamma}}{k}\right)^{k-k} g_i(T_i^k) + \left(1-\frac{1}{k}\right)^{k-k}\ell_i(T_i^k)\right)\\ &\qquad = \sum_{i=1}^{m}\left(g_i(T_i^k) + \ell_i(T_i^k)\right).\end{aligned}$$

Using Lemmas 1 and Lemma 3, we have

$$\begin{aligned}&\Phi_j(S^j)-\Phi_{j-1}(S^{j-1})\\ =&\sum_{i=1}^{m}\left(\nabla_i(t^j,T_i^{j-1})+\frac{\gamma+\frac{1}{\gamma}}{k}\left(1-\frac{\gamma+\frac{1}{\gamma}}{k}\right)^{k-j}g_i(T_i^{j-1})+\frac{1}{k}\left(1-\frac{1}{k}\right)^{k-j}\ell_i(T_i^{j-1})\right).\end{aligned}$$

Finally,

$$\begin{aligned}&\sum_{i=1}^{m}\left(g_i(T_i^k)+\ell_i(T_i^k)\right)\\ =&\sum_{j=1}^{k}(\Phi_j(S^j)-\Phi_{j-1}(S^{j-1}))\\ \geq&\sum_{j=1}^{k}\left(\frac{\gamma}{k}\left(1-\frac{\gamma+\frac{1}{\gamma}}{k}\right)^{k-j}\sum_{i=1}^{m}g_i(T_i^*)+\frac{1}{k}\left(1-\frac{1}{k}\right)^{k-j}\sum_{i=1}^{m}\ell_i(T_i^*)\right)\\ \geq&\frac{\gamma}{\gamma+\frac{1}{\gamma}}\left(1-e^{-(\gamma+\frac{1}{\gamma})}\right)\sum_{i=1}^{m}g_i(T_i^*)+\left(1-e^{-1}\right)\sum_{i=1}^{m}\ell_i(T_i^*).\end{aligned}$$

The curvature is an very useful assumption to obtain the following result.

Theorem 2. *There exists an algorithm returning a set S^k of size k such that*

$$F(S^k)\geq\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}\left(1-(1-\alpha)\gamma\right)(1-e^{-(\gamma+\frac{1}{\gamma})})+(1-\alpha)\gamma(1-e^{-1})\right)OPT.$$

where $F(S^k)=\sum_{i=1}^{m}\max_{T\in\mathcal{I}(S^k)}(f_i(T_i^k))$ and OPT is the optimal solution.

Proof. It follows from the definition of $\ell_i(T)$ and Proposition 2 that

$$\ell_i(T)=(1-\alpha)\sum_{v\in T}f_i(v)\geq(1-\alpha)\gamma f_i(T).$$

Furthermore,

$$\begin{aligned}&\sum_{i=1}^{m}f_i(T_i^k)=\sum_{i=1}^{m}\left(g_i(T_i^k)+\ell_i(T_i^k)\right)\\ \geq&\frac{\gamma}{\gamma+\frac{1}{\gamma}}\left(1-e^{-(\gamma+\frac{1}{\gamma})}\right)\sum_{i=1}^{m}g_i(T_i^*)+\left(1-e^{-1}\right)\sum_{i=1}^{m}\ell_i(T_i^*)\\ =&\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}(1-e^{-(\gamma+\frac{1}{\gamma})})\right)\sum_{i=1}^{m}(f_i(T_i^*)-\ell_i(T_i^*))+(1-e^{-1})\sum_{i=1}^{m}\ell_i(T_i^*)\\ =&\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}(1-e^{-(\gamma+\frac{1}{\gamma})})\right)\sum_{i=1}^{m}f_i(T_i^*)+\left((1-e^{-1})-\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}(1-e^{-(\gamma+\frac{1}{\gamma})})\right)\right)\sum_{i=1}^{m}\ell_i(T_i^*)\\ \geq&\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}(1-e^{-(\gamma+\frac{1}{\gamma})})\right)\sum_{i=1}^{m}f_i(T_i^*)+(1-\alpha)\gamma\left((1-e^{-1})-\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}(1-e^{-(\gamma+\frac{1}{\gamma})})\right)\right)\sum_{i=1}^{m}f_i(T_i^*)\\ \geq&\left(\frac{\gamma}{\gamma+\frac{1}{\gamma}}\left(1-(1-\alpha)\gamma\right)(1-e^{-(\gamma+\frac{1}{\gamma})})+(1-\alpha)\gamma(1-e^{-1})\right)\sum_{i=1}^{m}f_i(T_i^*).\end{aligned}$$

Finally, we obtain that

$$F(S^k) \geq \left(\frac{\gamma}{\gamma + \frac{1}{\gamma}} (1 - (1-\alpha)\gamma) (1 - e^{-(\gamma + \frac{1}{\gamma})}) + (1-\alpha)\gamma(1 - e^{-1}) \right) OPT.$$

4 Conclusion

The objective functions for many applications in practice are in general not submodular. To solve these optimization problems, an important research method is to introduce some parameters to describe the characteristics of the non-submodular functions, such as submodularity ratio, curvature, supermodular degree, etc., and then design algorithms for the problems and analyze the performances of the algorithms with these parameters. On the other hand, it is well known that submodular maximization problem can be solved by greedy algorithms, To avoid this limitation of the regular greedy algorithm, we propose combining the distorted objective function and the greedy algorithms, which has the potential to be applicable to other optimization problems.

Acknowledgements. The research is supported by NSFC (Nos.12101314,12131003, 11871280,12271259,11971349), Qinglan Project, Natural Science Foundation of Jiangsu Province (No. BK20200723), and Jiangsu Province Higher Education Foundation (No.20KJB110022).

References

1. Balkanski, E., Krause, A., Mirzasoleiman, B., Singer, Y.: Learning sparse combinatorial representations via two-stage submodular maximization. In: ICML, pp. 2207–2216 (2016)
2. Bian, A.A., Buhmann, J.M., Krause, A., Tschiatschek, S.: guarantees for greedy maximization of non-submodular functions with applications. In: Proceedings of ICML, pp. 498–507 (2017)
3. Conforti, M., Cornuejols, G.: Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. Discrete Appl. Math. **7**(3), 251–274 (1984)
4. Gong, S., Nong, Q., Sun, T., Fang, Q., Du, X., Shao, X.: Maximize a monotone function with a generic submodularity ratio. Theor. Comput. Sci. **853**, 16–24 (2021)
5. Krause, A., Guestrin, A.: Near-optimal nonmyopic value of information in graphical models. In: UAI, p. 5 (2005)
6. Laitila, J., Moilanen, A.: New performance guarantees for the greedy maximization of submodular set functions. Optim. Lett. **11**, 655–665 (2017)
7. Lee, J., Mirrokni, V.S., Nagarajan, V., Sviridenko, M.: Maximizing nonmonotone submodular functions under matroid or knapsack constraints. SIAM J. Discrete Math. **23**(4), 2053–2078 (2010)
8. Lawrence, N., Seeger, M., Herbrich, R.: Fast sparse gaussian process methods: the informative vector machine. Adv. Neural. Inf. Process. Syst. **1**(5), 625–632 (2003)

9. Mitrovic, M., Kazemi, E., Zadimoghaddam, M., Karbasi, A.: Data summarization at scale: a two-stage submodular approach. In: ICML, pp. 3593–3602 (2018)
10. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-i. Math. Program. **14**(1), 265–294 (1978)
11. Schulz, A.S., Uhan, N.A.: Approximating the least core value and least core of cooperative games with supermodular costs. Discrete Optim. **10**(2), 163–180 (2013)
12. Stan, S., Zadimoghaddam, M., Krause, A., Karbasi, A.: Probabilistic submodular maximization in sub-linear time. In: ICML, pp. 3241–3250 (2017)
13. Yang, R., Gu, S., Gao, C., Wu, W., Wang, H., Xu, D.: A constrained two-stage submodular maximization. Theor. Comput. Sci. **853**, 57–64 (2021)