SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
April 2022, Vol. 65 140602:1–140602:10
https://doi.org/10.1007/s11432-020-3420-9
Special Focus on Theory and Applications of Models of Computation

Regularized two-stage submodular maximization under streaming

Ruiqi YANG[1,2], Dachuan XU[1], Longkun GUO[3] & Dongmei ZHANG[4*]

[1]*Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing* 100124*, China;*
[2]*School of Mathematical Sciences, University of Chinese Academy Sciences, Beijing* 100049*, China;*
[3]*School of Computer Science and Technology, Qilu University of Technology, Jinan* 250353*, China;*
[4]*School of Computer Science and Technology, Shandong Jianzhu University, Jinan* 250101*, China*

Received 31 December 2020/Revised 15 October 2021/Accepted 1 December 2021/Published online 14 March 2022

Abstract In the problem of maximizing regularized two-stage submodular functions in streams, we assemble a family $\mathcal{F}$ of m functions each of which is submodular and is visited in a streaming style that an element is visited for only once. The aim is to choose a subset S of size at most ℓ from the element stream $\mathcal{V}$, so as to maximize the average maximum value of these functions restricted on S with a regularized modular term. The problem can be formally casted as $\max_{S\subseteq V,|S|\leqslant\ell}\frac{1}{m}\sum_{i=1}^{m}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]$, where $c:\mathcal{V}\to\mathbb{R}_+$ is a non-negative modular function and $f_i:2^{\mathcal{V}}\to\mathbb{R}_+,\forall i\in\{1,\ldots,m\}$ is a non-negative monotone non-decreasing submodular function. The well-studied regularized problem of $\max_{S\subseteq\mathcal{V},|S|\leqslant k}f(S)-c(S)$ is exactly a special case of the above regularized two-stage submodular maximization by setting $m=1$ and $\ell=k$. Although $f(\cdot)-c(\cdot)$ is submodular, it is potentially negative and non-monotone and admits no constant multiplicative factor approximation. Therefore, we adopt a slightly weaker notion of approximation which constructs S such that $f(S)-c(S)\geqslant\rho\cdot f(O)-c(O)$ holds against optimum solution O for some $\rho\in(0,1)$. Eventually, we devise a streaming algorithm by employing the distorted threshold technique, achieving a weaker approximation ratio with $\rho=0.2996$ for the discussed regularized two-stage model.

Keywords submodular maximization, streaming model, two-stage, threshold technique, approximation algorithms

Citation Yang R Q, Xu D C, Guo L K, et al. Regularized two-stage submodular maximization under streaming. Sci China Inf Sci, 2022, 65(4): 140602, https://doi.org/10.1007/s11432-020-3420-9

1 Introduction

Summarization techniques designated for large amounts of data (elements) have piqued the interest of researchers in computer science, machine learning, combinatorial optimization, and many other fields. In general, most summarization tasks can be reduced into sieving a subset of elements with the aim of maximizing a utility function that quantifies the "representativeness" of the chosen set. Further, in most cases these utility functions satisfy intuitive diminishing returns which can be formally defined as submodularity. Submodular maximization has received extensive research due to its significant theoretical implications, as well as broad applications such as influence maximization [1], document summarization [2], network monitor [3].

Consider an example of image summarization. Assume that there are some images that can be partitioned into different categories according to different attributes of their contexts. The aim is to choose a subset that accurately expresses each of these attributes. Mathematically, the above example can be intuitively described as a two-stage submodular maximization problem introduced by [4]. Letting $\mathcal{V}$ denote an element ground set, a collection $\mathcal{F}$ consists of m submodular functions, the goal is to choose a subset S of size bounded by ℓ, with the average value of these functions restricted on S being as large as possible. We restate the problem as

$$\max_{S\subseteq\mathcal{V},|S|\leqslant k}\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}f_i(T). \tag{1}$$

* Corresponding author (email: zhangdongmei@sdjzu.edu.cn)

Growing modern datasets and the rapidly produced data bring challenges to the storage capacity of the storage resource. Hence, the streaming model is popular as it can elegantly deal with submodular maximization applications on large datasets. In streaming fashion, assume we only have access to a small fraction of the data stored in the primary memory, while the elements are revealed one by one, the performance guarantees of a streaming algorithm can be formally measured by four parameters introduced by [5] (1) the number of passes produced by the algorithm, (2) the memory complexity, (3) the update time (i.e., the number of oracle queries), and (4) the approximation ratio.

1.1 Our contributions

We study the two-stage submodular maximization model under a streaming fashion. In addition, we observe that it is better to select any given element earlier rather than later, which may result in the over-fitting phenomenon in our selection process. To deal with the hardness, we use the two-stage submodular maximization by adding a non-decreasing function $c : V \to \mathbb{R}_+$ as the regularized term. We define the regularized problem as

$$\max_{S\subseteq\mathcal{V},|S|\leqslant k}\left\{\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]\right\}. \tag{2}$$

Furthermore, even if $m = 1$ and $\ell = k$, the two-stage submodular maximization problem of maximizing $\max_{S\subseteq\mathcal{V},|S|\leqslant k} f(T) - c(T)$ remains inapproximable, meaning that no multiplicative factor approximation is possible unless $P = NP$. Therefore, researchers focus on a slightly weaker notion of approximation, and aim to design algorithms that construct a solution $S \subseteq \mathcal{V}$ satisfying

$$f(S) - c(S) \geqslant \rho \cdot f(O) - c(O)$$

for some $\rho < 1$ and any optimum solution $O = \arg\max_{S\subseteq\mathcal{V},|S|\leqslant k} f(S) - c(S)$. For the streaming regularized two-stage submodular maximization, we introduce a distorted threshold procedure which can eventually obtain a subset S such that

$$\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)] \geqslant \frac{1}{m}\sum_{i\in[m]}[0.2996\cdot f_i(S_i^*)-c(S_i^*)], \tag{3}$$

where $S^* \in \arg\max_{S\subseteq\mathcal{V},|S|\leqslant\ell}\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]$ represents the optimum solution and $S_i^* \in \arg\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]$ represents the optimum solution with respect to function f_i.

1.2 Related work

The constrained submodular maximization is usually cast as $\max_{S\subseteq\mathcal{V},S\in\mathcal{I}} f(S)$, where $\mathcal{I}$ is a specified constraint such as cardinality, knapsack, matroid constraint. For the submodular maximization with a cardinality constraint, Nemhauser et al. [6] presented a greedy-based $(1-e^{-1})$-approximation algorithm, which chooses an element with a maximum marginal gain during any iteration. In addition, the submodular maximization with a knapsack or matroid constraint can also be optimized within a factor of $(1-e^{-1})$ as summarized by [7, 8].

The streaming-based model is one of the most popular approaches for addressing optimization applications with large-scale or real-time generated elements (data). The elements are read sequentially in the model, and we only have access to a limited amount of memory at any given time during the procession.

For the streaming scenario of cardinality constrained submodular maximization, Badanidiyuru et al. [5] provided a threshold-based $(0.5-\varepsilon)$-approximation, which makes one pass over the stream, consumes $O(\varepsilon^{-1}k\log k)$ memory and needs $O(\varepsilon^{-1}\log k)$ update time per element. Buchbinder et al. [9] later yielded a 0.25-approximation algorithm with $O(k)$ memory and $O(nk)$ update time. Assume the elements are presented in a random order, Norouzi-Fard et al. [10] gave a $(0.5+10^{-14})$-approximation. Moreover, they show there exists no $(0.5+\varepsilon)$-approximation algorithm with a memory complexity less than $O(n/k)$ if the elements are revealed in an arbitrary order. Kazemi et al. [11] provided an improved algorithm, which decreases the memory complexity from $O(\varepsilon^{-1}k\log k)$ to $O(\varepsilon^{-1}k)$, but retains the other performance

guarantees at the same time. More work about streaming algorithms for submodular maximization with complex constraints was investigated by [12–16].

The regularized submodular maximization is stated as $\arg\max_{S\subseteq\mathcal{V},|S|\leqslant k} f(S)-c(S)$, where f and c are monotone non-decreasing and non-negative modular functions, respectively. Note that the regularized model is inapproximable, since the objective function value may be negative and non-monotone. Researchers use a slightly weaker notion of approximation to measure the performance guarantees, which is to choose a subset S, such that $f(S)-c(S)\geqslant\rho\cdot f(S^*)-c(S^*), \rho<1$. Based on the greedy approach, Refs. [17, 18] provided algorithms with approximation ρ near to $1-e^{-1}$. Further, Harshaw et al. [19] provided a distorted greedy method, which iteratively and greedily finds an element with a maximized distorted marginal gain. Combining with a submodular ratio γ, they derive a weak approximation with $\rho=1-e^{-\gamma}$. Kazemi et al. [20] studied the regularized model under streaming and presented a distorted threshold-based algorithm with $\rho=0.382$.

Balkanski et al. [4] introduced the two-stage submodular maximization model and provided a $(1-1/e)/2$-approximation algorithm with $O(\ell mn^2\log(n)k)$ time complexity. Stan et al. [21] then derived a $(1-1/e^2)/2$-approximation, which has an improved time complexity of $O(\ell mnk)$. Considering the two-stage model under streaming scenario, Mitrovic et al. [22] provided a $1/(6+\varepsilon)$-approximation streaming algorithm, which needs $O(\varepsilon^{-1}\ell\log\ell)$ memory and has $O(\varepsilon^{-1}kmn\log\ell)$ update time. Based on a parameter of generalized submodularity ratio introduced by [23], we develop a parameterized streaming algorithm for the two-stage submodular maximization appeared in [24].

Organization. The remainder of the paper is organized as follows. Preliminaries are given in Section 2. The main streaming algorithm is presented in Section 3 and its theoretical analysis is postponed in Section 4. Lastly, Section 5 concludes the study.

2 Preliminaries

Letting $\mathcal{V}$ denote an element ground set, we say a set function $f:2^{\mathcal{V}}\to\mathbb{R}$ is submodular, if and only if for any two subsets $A,B\subseteq\mathcal{V}$ we have

$$f(A)+f(B)\geqslant f(A\cup B)+f(A\cap B).$$

Let $f(x|A)=f(A+e)-f(A)$ be the marginal gain of adding element e to subset A. An equivalent form of submodularity over set functions can be stated as

$$f(x|A)\geqslant f(x|B),\quad \forall A\subseteq B\subseteq\mathcal{V},\quad x\notin B.$$

The submodular function f is monotone and non-decreasing if $f(x|A)\geqslant 0$ for any $A\subseteq\mathcal{V}, x\in\mathcal{V}$. We set $[n]=\{1,\ldots,n\}$ for any integer n. A more powerful expression of submodular maximization is stated as

$$\max_{S\subseteq\mathcal{V},|S|\leqslant k}\left\{\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k} f_i(T)\right\}.$$

In this model, assume we assemble a collection $\mathcal{F}$ of m set functions, where each function $f:2^{\mathcal{V}}\to\mathbb{R}_+$ is monotone non-decreasing and submodular. The goal is to choose a subset with size at most ℓ, so as to maximize the average optimal value of the functions f_i, $i\in[m]$ restricted on S.

We study the above two-stage model with two twists. Assume the elements are revealed in a streaming fashion. We further add a regularized non-decreasing modular function $c:\mathcal{V}\to\mathbb{R}_+$ to balance the chosen process. Formally, we define the regularized two-stage submodular maximization problem as

$$\max_{S\subseteq\mathcal{V},|S|\leqslant\ell} G(S)=\max_{S\subseteq\mathcal{V},|S|\leqslant\ell}\left\{\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]\right\},$$

where $f_i,\forall i\in[m]$ is monotone non-decreasing submodular and c is modular.

In addition, we assume there is an oracle for computing the submodular utility for any subsets, such that we can query $f(S)$ in time of $O(1)$ for any $S\subseteq\mathcal{V}$.

3 Threshold-based streaming algorithm

We provide a threshold-based streaming algorithm in this section. The main idea behind our distorted algorithm is to yield a subset S of size at most ℓ from the stream by carefully selecting a threshold based on a distorted optimum value. Further, by introducing distorted add and swap operations, we iteratively update subsets $\{T_i\}_i \subseteq S, \forall i \in [m]$ to influence the selection of the next element. The main pseudo codes are listed in Algorithm 1.

Let $H(\alpha,\lambda) = \frac{(\lambda-1)\alpha}{(\alpha+1)\lambda+(\lambda-1)(\alpha+2)\alpha}$, where $\alpha > 0$ and $\lambda > 1$ are to be determined in Section 4. Assume we have access to the threshold τ^*, such that

$$\tau^* = \frac{1}{\ell}\left\{\frac{H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\cdot\frac{1}{m}\sum_{i\in[m]} f_i(S_i^*) - \lambda c(S_i^*)\right\}, \tag{4}$$

where $S^* \in \arg\max_{S\subseteq\mathcal{V},|S|\leqslant\ell}\frac{1}{m}\sum_{i\in[m]}\max_{T\subseteq S,|T|\leqslant k}[f_i(T)-c(T)]$ represents the optimum solution and $S_i^* \in \arg\max_{T\subseteq S^*,|T|\leqslant k}[f_i(T)-c(T)]$ represents an optimum solution with respect to function f_i, for any $i \in [m]$.

Next, we introduce the new distorted add and swap operations. For any subset $A \subseteq \mathcal{V}$ and element $x \in \mathcal{V}$, $\lambda > 0$, let

$$\Delta_i^a(x|A) = [f_i(A+x)-\lambda c(A+x)] - [f_i(A)-\lambda c(A)] = f_i(A+x) - f_i(A) - \lambda c(x)$$

denote the distorted gain of adding element x to A according to f_i and regularized term c. Moreover, let

$$\mathrm{Rep}_i(x,A) = \arg\max_{y\in A}\{[f_i(A+x-y)-\lambda c(A+x-y)]-[f_i(A)-\lambda c(A)]\} = f_i(A+x-y)-f_i(A)+\lambda(c(y)-c(x))$$

denote an element of A which is swapped by x with the largest gain according to f_i and c. The distorted swapped gain is defined as

$$\begin{aligned}\Delta_i^e(x,A) &= [f_i(A+x-\mathrm{Rep}_i(x,A))-\lambda c(A+x-\mathrm{Rep}_i(x,A))]-[f_i(A)-\lambda c(A)]\\ &= f_i(A+x-\mathrm{Rep}_i(x,A)) - f_i(A) + \lambda(c(\mathrm{Rep}_i(x,A))-c(x)).\end{aligned}$$

Define the gain of element x with respect to A as

$$\nabla_i(x,A) = \begin{cases}\mathbf{1}_{\{\Delta_i^a(x|A)\geqslant\frac{\alpha}{k}\cdot(f_i(A)-\lambda c(A))\}}\cdot\Delta_i^a(x|A), & \text{if } |A|<k,\\ \mathbf{1}_{\{\Delta_i^e(x|A)\geqslant\frac{\alpha}{k}\cdot(f_i(A)-\lambda c(A))\}}\cdot\Delta_i^e(x|A), & \text{otherwise,}\end{cases}$$

where $\mathbf{1}$ represents the indicator function, and α and λ are the tunable parameters.

The procedure starts with $S = \emptyset$ and $T_i = \emptyset, \forall i \in [m]$. Considering the visiting element e_t at the current time t, if $|S|$ is less than ℓ and the average gain of element e_t according to T^{t-1} is greater than or equal to the threshold value τ^*, we add the visited element e_t to S. For each $i \in [m]$, we discuss the following two cases to update the set of T_i^t. For the subcase of $\nabla_i(e_t, T_i^{t-1}) > 0$ and $|T_i^{t-1}| < k$, we implement the add operation, i.e., $T_i^t = T_i^{t-1} + e_t$. Otherwise, we execute the swap operation, i.e., $T_i^t = T_i^{t-1} + e_t - \mathrm{Rep}_i(e_t, T_i^{t-1})$. The procedure terminates once the stream ends or the cardinality constraint is saturated. Since the optimum threshold value τ^* is not known a priori, we implement the procedure by adding the guessing steps as stated by lines 3 and 4 in Algorithm 1.

4 Theoretical analysis

Let S^t denote the set of retained elements till time t. For each function f_i, we represent T_i^t as the set of kept elements till time t. Let $A_i^t = \cup_{t'\leqslant t} T_i^{t'}$ denote the set of elements which ever has appeared in T_i until time t. Considering the first k add operations to set T_i^t, one readily has $T_i^t = A_i^t$ and $f_i(T_i^t) = f_i(A_i^t)$ for any $i \in [m]$. After the first k add operations, consider the arrived element e_t. The algorithm performs the swap operation by adding e_t to T_i^t and simultaneously deleting $\mathrm{Rep}_i(e_t, T_i^t)$ from T_i^t. Then we get the following lemma.

Algorithm 1 Distorted-replacement-streaming

Input: Elements stream $V = \{e_1, e_2, \ldots\}$, $H(\alpha, \lambda) = \frac{(\lambda-1)\alpha}{(\alpha+1)\lambda+(\lambda-1)(\alpha+2)\alpha}$.
Ensure: Summarization S; Surrogate sets $\{T_i\}_{i=1}^m$.
Initialization $S \leftarrow \emptyset$, $T_i \leftarrow \emptyset, \forall i \in [m]$, $t \leftarrow 1, \delta_0 \leftarrow 0$;
while $|S| < \ell$ and a new element e_t reveals **do**
  $\delta_t \leftarrow \max\{\delta^{t-1}, \frac{1}{m}\sum_{i\in[m]} f_i(e_t) - \lambda c(e_t)\}$,
  $O_t \leftarrow \{(1+\varepsilon)^i | \frac{(1-\varepsilon)\delta_t H(\alpha,\lambda)}{\ell(H(\alpha,\lambda)+1)} \leqslant (1+\varepsilon)^i \leqslant \delta_t\}$,
  for each $\tau \in O_t$ **do**
    if τ is a new instantiated value **then**
      $S \leftarrow \emptyset$, $T_i^t \leftarrow \emptyset$;
    end if
    if $\frac{1}{m}\sum_{i\in[m]} \nabla_i(e_t, T_i^{t-1}) \geqslant \tau$ **then**
      $S \leftarrow S + e_t$;
      for any $i \in [m]$ **do**
        Case 1. $\nabla_i(e_t, T_i^{t-1}) > 0 \wedge |T_i^{t-1}| < K$, $T_i^t \leftarrow T_i^{t-1} + e_t$;
        Case 2. $\nabla_i(e_t, T_i^{t-1}) > 0 \wedge |T_i^{t-1}| = K$, $T_i^t \leftarrow T_i^{t-1} + e_t - \mathrm{Rep}_i(e_t, T_i^{t-1})$;
      end for
    end if
  end for
  $t \leftarrow t + 1$.
end while

Lemma 1. For any $i \in [m]$, we have

$$\Delta_i^e(e_t, T_i^{t-1}) \geqslant [f_i(e_t|T_i^{t-1}) - \lambda c(e_t)] - \frac{1}{k} \cdot [f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})].$$

Proof. To show the above claim, we start from the following inequalities:

$$\begin{aligned}
\Delta_i^e(e_t, T_i^{t-1}) =\ & f_i(T_i^{t-1} + e_t - \mathrm{Rep}_i(e_t, T_i^{t-1})) - f_i(T_i^{t-1}) + \lambda(c(\mathrm{Rep}_i(e_t, T_i^{t-1})) - c(e_t)) \\
\geqslant\ & \frac{1}{k} \cdot \sum_{e\in T_i^{t-1}} f_i(T_i^{t-1} + e_t - e) - f_i(T_i^{t-1}) + \lambda(c(e) - c(e_t)) \\
\geqslant\ & \frac{1}{k} \cdot \sum_{e\in T_i^{t-1}} [f_i(T_i^{t-1} + e_t) - f_i(T_i^{t-1})] + [f_i(T_i^{t-1} - e) - f_i(T_i^{t-1})] \\
& -\frac{\lambda}{k} \cdot \sum_{e\in T_i^{t-1}} [c(T_i^{t-1} + e_t) - c(T_i^{t-1})] + [c(T_i^{t-1} - e) - c(T_i^{t-1})] \\
\geqslant\ & [f_i(e_t|T_i^{t-1}) - \lambda c(e_t)] - \frac{1}{k} \cdot [f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})].
\end{aligned}$$

The first inequality is obtained by the definition of $\mathrm{Rep}_i(e_t, T_i^t)$ and the followed inequalities are obtained by submodularity.

Based on the above lemma, we get a lower bound of $f_i(T_i^t) - \lambda c(T_i^t)$ by calculating $f_i(A_i^t) - \lambda c(T_i^t)$.

Lemma 2. For any $i \in [m], \alpha, \lambda > 0$, we have

$$f_i(T_i^t) - \lambda c(T_i^t) \geqslant \frac{\alpha}{\alpha+1} \cdot (f_i(A_i^t) - \lambda c(T_i^t)).$$

Proof. The proof completes by the induction of iterations. We assume the lemma corrects at the step of $t-1$, i.e.,

$$f_i(T_i^{t-1}) - \lambda c(T_i^{t-1}) \geqslant \frac{\alpha}{\alpha+1} \cdot (f_i(A_i^{t-1}) - \lambda c(A_i^{t-1})).$$

By the definition of $\Delta_i^e(e_t, T_i^{t-1})$, we have $\Delta_i^e(e_t, T_i^{t-1}) \geqslant \frac{\alpha}{k} \cdot (f_i(T_i^t) - \lambda c(T_i^t))$. Then combining with Lemma 1, we derive

$$\begin{aligned}
[f_i(T_i^t) - \lambda c(T_i^t)] - [f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})] \geqslant \max\Bigg\{ & \frac{\alpha}{k} \cdot (f_i(T_i^t) - \lambda c(T_i^t)), \\
& [f_i(e_t|T_i^{t-1}) - \lambda c(e_t)] - \frac{1}{k} \cdot [f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})] \Bigg\}
\end{aligned}$$

$$\begin{aligned}&\geqslant \frac{\alpha}{\alpha+1}\cdot[f_i(e_t|A_i^{t-1})-\lambda c(e_t)]\\&= \frac{\alpha}{\alpha+1}\cdot[f_i(A_i^t)-\lambda c(A_i^t)]-[f_i(A_i^{t-1})-\lambda c(A_i^{t-1})].\end{aligned}$$

Then by the induction hypothesis, we get

$$f_i(T_i^t)-\lambda c(T_i^t)\geqslant \frac{\alpha}{\alpha+1}\cdot(f_i(A_i^t)-\lambda c(A_i^t)).$$

We describe the ideas on how to efficiently estimate the distorted threshold value of τ^*. Although the optimal threshold value τ^* depends on the distorted optimum value, which is not exactly known in advance, we use the following lemma to guess the distorted optimum value.

Lemma 3. Let $\delta=\frac{1}{m}\sum_{i\in[m]}\max_{e\in\mathcal{V}}[f_i(e)-\lambda c(e)]$ and $\lambda>1$. We have the following inequalities:

$$\frac{\delta H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\leqslant\frac{H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\left\{\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}\leqslant\frac{\delta\ell H(\alpha,\lambda)}{H(\alpha,\lambda)+1}.$$

Proof. Since any $e\in\mathcal{V}$ is feasible, we have

$$\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\geqslant\frac{1}{m}\sum_{i\in[m]}\max_{e\in\mathcal{V}}[f_i(e)-\lambda c(e)]=\delta.$$

By submodularity, we have

$$\frac{1}{m}\sum_{i\in[m]}[f_i(S_i^*)-\lambda c(S_i^*)]\leqslant\frac{1}{m}\sum_{i\in[m]}\sum_{e\in S_i^*}[f_i(e)-\lambda c(e)]\leqslant\ell\delta.$$

Based on the above lemma, we can construct

$$O=\left\{(1+\varepsilon)^i\middle|\frac{(1-\varepsilon)\delta H(\alpha,\lambda)}{\ell(H(\alpha,\lambda)+1)}\leqslant(1+\varepsilon)^i\leqslant\frac{\delta H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\right\}$$

and guess $\tau\in O$ in a runtime of $O(\varepsilon^{-1}\log\ell)$ such that $(1-\varepsilon)\tau^*\leqslant\tau\leqslant\tau^*$. However, the maximum singleton value $\delta=\frac{1}{m}\max_{e\in\mathcal{V}}[f_i(e)-\lambda c(e)]$ only can be learned after the stream ends. So we guess the optimal τ^* approximately by the the value of δ_t, which denotes the singleton value at the current time t. Before presenting the approximate guess process, we obtain the next lemma.

Lemma 4. Let $\delta_t=\frac{1}{m}\max_{e_{t'},t'\leqslant t}\sum_{i\in[m]}f_i(e_{t'})-\lambda c(e_{t'})$ denote the maximum singleton value at the time of step t. Then we have

$$\frac{1}{m}\sum_{i\in[m]}\nabla_i(e_t,T_i^{t-1})\leqslant\delta_t.$$

Proof. To prove this claim we have the following:

$$\frac{1}{m}\sum_{i\in[m]}\nabla_i(e_t,T_i^{t-1})\leqslant\frac{1}{m}\sum_{i\in[m]}\Delta_i^a(e_t,T_i^{t-1})\leqslant\frac{1}{m}\sum_{i\in[m]}f_i(e_t)-\lambda c(e_t)\leqslant\delta_t.$$

Now we construct

$$O_t=\left\{(1+\varepsilon)^i\middle|\frac{(1-\varepsilon)\delta_t H(\alpha,\lambda)}{\ell(H(\alpha,\lambda)+1)}\leqslant(1+\varepsilon)^i\leqslant\delta_t\right\}$$

and also can guess $\tau\in\cup_t O_t$ with $(1-\varepsilon)\tau^*\leqslant\tau\leqslant\tau^*$. Equipped with the above lemmas, we get our main result by the following theorem.

Theorem 1. For any $\epsilon>0$ and $\alpha>0,\lambda>1$, Algorithm 1 returns a subset S satisfying

$$G(S)\geqslant\frac{H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\cdot\left\{\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}.$$

Proof. Let S and $\{T_i\}_{i=1}^m$ denote the returned sets of Algorithm 1 according to threshold $\tau \in \cup_t O_t$ subject to $(1-\epsilon)\tau^* \leqslant \tau \leqslant \tau^*$. Our discussion consists of two cases: $|S| = \ell$ and $|S| < \ell$. For the case of $|S| = \ell$, we obtain

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^m f_i(T_i^n) - \lambda c(T_i^n) &= \frac{1}{m}\sum_{t=1}^n\sum_{i=1}^m [f_i(T_i^t) - \lambda c(T_i^t)] - [f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})] \\ &= \frac{1}{m}\sum_{i=1}^m \mathbf{1}_{e_t\in S}\cdot \nabla_i(e_t, T_i^{t-1}) \\ &\geqslant \ell\tau.\end{aligned}$$

Setting $\lambda > 1$, then

$$\frac{1}{m}\sum_{i=1}^m f_i(T_i^n) - c(T_i^n) \geqslant \frac{1}{m}\sum_{i=1}^m f_i(T_i^n) - \lambda c(T_i^n) \geqslant \ell\tau. \tag{5}$$

For the case of $|S| < \ell$, fixing any $i \in [m]$, we partition $S_i^* \setminus A_i^n$ into the following three parts. Let D_i denote the set of elements of $S_i^* \setminus A_i^n$ encountered before setting threshold τ and $D = \cup_{i\in[m]} D_i$. The other elements of $S_i^* \setminus A_i^n$ are partitioned into

$$P_i := \left\{e_t \in S_i^* \setminus A_i^n | \Delta_i^a(e_t, T_i^{t-1}) < \frac{\alpha}{k}\cdot(f_i(T_i^{t-1}) - \lambda c(T_i^{t-1}))\right\}$$

and

$$Q_i := \left\{e_t \in S_i^* \setminus A_i^n | \Delta_i^a(e_t, T_i^{t-1}) \geqslant \frac{\alpha}{k}\cdot(f_i(T_i^{t-1}) - \lambda c(T_i^{t-1})) \wedge \frac{1}{m}\sum_{i\in[m]} \nabla_i(e_t, T_i^{t-1}) < \tau\right\}.$$

Since

$$\begin{aligned}\frac{1}{m}\sum_{i\in[m]} f_i(S_i^*) - f_i(A_i^n) - \lambda c(S_i^*) &\leqslant \frac{1}{m}\sum_{i\in[m]} f_i(S_i^*\cup A_i^n) - f_i(A_i^n) - \lambda(c(S_i^*\cup A_i^n) - c(A_i^n)) \\ &\leqslant \frac{1}{m}\sum_{i\in[m]}\sum_{e_t\in S_i^*\setminus A_i^n} f_i(e_t|A_i^n) - \lambda c(e_t) \\ &\leqslant \frac{1}{m}\sum_{i\in[m]}\sum_{e_t\in S^*} \mathbf{1}_{e_t\in S_i^*}\{\mathbf{1}_{e_t\in D_i}[f_i(e_t|A_i^n) - \lambda c(e_t)] \\ &\quad +\mathbf{1}_{e_t\notin D_i}\mathbf{1}_{e_t\in P_i}[f_i(e_t|A_i^n) - \lambda c(e_t)] \\ &\quad +\mathbf{1}_{e_t\notin D_i}\mathbf{1}_{e_t\in Q_i}[f_i(e_t|A_i^n) - \lambda c(e_t)]\}.\end{aligned} \tag{6}$$

Considering elements in D_i, we get

$$\frac{1}{m}\sum_{i\in[m]} \nabla_i(e_t, T_i^{t-1}) \leqslant \frac{1}{m}\sum_{i\in[m]} \Delta_i^a(e_t|T_i^{t-1}) \leqslant \frac{1}{m}\sum_{i\in[m]} f_i(e_t) - \lambda c(e_t) \leqslant \delta_t < \tau.$$

Then the first term of inequality (6) can be bounded by

$$\frac{1}{m}\sum_{i\in[m]}\sum_{e_t\in S^*} \mathbf{1}_{e_t\in S_i^*}\mathbf{1}_{e_t\in D_i}[f_i(e_t|A_i^n) - \lambda c(e_t)] \leqslant |D|\tau. \tag{7}$$

Next we consider elements in P_i. Since element $e_t \in P_i$, we have $\Delta_i^a(e_t, T_i^{t-1}) < \frac{\alpha}{k}\cdot(f_i(T_i^{t-1}) - \lambda c(T_i^{t-1}))$ and $f_i(e_t|A_i^n) - \lambda c(e_t) \leqslant \frac{\alpha+1}{k}\cdot[f_i(e_t|T_i^n) - \lambda c(T_i^n)]$. Then

$$\frac{1}{m}\sum_{i\in[m]}\sum_{e_t\in S^*} \mathbf{1}_{e_t\in S_i^*}\mathbf{1}_{e_t\notin D_i}\mathbf{1}_{e_t\in P_i}[f_i(e_t|A_i^n) - \lambda c(e_t)] \leqslant (\alpha+1)\cdot\frac{1}{m}\sum_{i\in[m]}[f_i(T_i^n) - \lambda c(T_i^n)]. \tag{8}$$

Following Lemma 1, we obtain the following for elements in Q_i:

$$\frac{1}{m}\sum_{i\in[m]}\mathbf{1}_{e_t\in S_i^*}\mathbf{1}_{e_t\in Q_i}\left[f_i(e_t|T_i^{t-1})-\lambda c(e_t)-\frac{1}{k}\cdot[f_i(T_i^{t-1})-\lambda c(T_i^{t-1})]\right]\leqslant\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}_{e_t\in S_i^*}\nabla_i(e_t,T_i^{t-1})\leqslant\tau.$$

Then we get

$$\frac{1}{m}\sum_{i\in[m]}\mathbf{1}_{e_t\in S_i^*}\mathbf{1}_{e_t\in Q_i}[f_i(e_t|A_i^{t-1})-\lambda c(e_t)]\leqslant\tau+\frac{1}{km}\sum_{i\in[m]}\mathbf{1}_{e_t\in S_i^*}[f_i(T_i^{t-1})-\lambda c(T_i^{t-1})].$$

Thus, we bound the second term of inequality (6) as

$$\frac{1}{m}\sum_{i\in[m]}\sum_{e_t\in S^*}\mathbf{1}_{e_t\in S_i^*}\mathbf{1}_{e_t\notin D_i}\mathbf{1}_{e_t\in Q_i}[f_i(e_t|A_i^{t-1})-\lambda c(e_t)]\leqslant(\ell-|D|)\tau+\frac{1}{m}\sum_{i\in[m]}[f_i(T_i^{t-1})-\lambda c(T_i^{t-1})]. \quad (9)$$

Based on inequalities (7)–(9), we get

$$\frac{1}{m}\sum_{i\in[m]}f_i(A_i^n)\geqslant-\ell\tau+\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)-(\alpha+2)\cdot\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-\lambda c(T_i^n).$$

Since we have $f_i(A_i^n)-\lambda c(A_i^n)\geqslant 0$ for any $i\in[m]$, combining the two inequalities with coefficients $(\lambda-1)/\lambda$ and $1/\lambda$ gives

$$\frac{1}{m}\sum_{i\in[m]}f_i(A_i^n)-\lambda c(A_i^n)\geqslant\frac{\lambda-1}{\lambda}\cdot\left\{-\ell\tau+\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)-(\alpha+2)\cdot\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-\lambda c(T_i^n)\right\}.$$

Following Lemma 2 and the fact of $\lambda>1$, we rearrange the above inequality,

$$\begin{aligned}\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-c(T_i^n)&\geqslant\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-\lambda c(T_i^n)\\&\geqslant\frac{(\lambda-1)\alpha}{(\alpha+1)\lambda+(\lambda-1)(\alpha+2)\alpha}\cdot\left\{-\ell\tau+\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}. \quad (10)\end{aligned}$$

Combining with inequalities (5) and (10), we obtain

$$\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-c(T_i^n)\geqslant\min\left\{\ell\tau,\frac{(\lambda-1)\alpha}{(\alpha+1)\lambda+(\lambda-1)(\alpha+2)\alpha}\cdot\left\{-\ell\tau+\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}\right\}.$$

Since $H(\alpha,\lambda)=\frac{(\lambda-1)\alpha}{(\alpha+1)\lambda+(\lambda-1)(\alpha+2)\alpha}$, by setting

$$\ell\tau=H(\alpha,\lambda)\cdot\left\{-\ell\tau+\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\},$$

we get

$$\ell\tau=\frac{H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\cdot\left\{\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}.$$

So eventually we have

$$G(S)\geqslant\frac{1}{m}\sum_{i\in[m]}f_i(T_i^n)-c(T_i^n)\geqslant\frac{H(\alpha,\lambda)}{H(\alpha,\lambda)+1}\cdot\left\{\frac{1}{m}\sum_{i\in[m]}f_i(S_i^*)-\lambda c(S_i^*)\right\}.$$

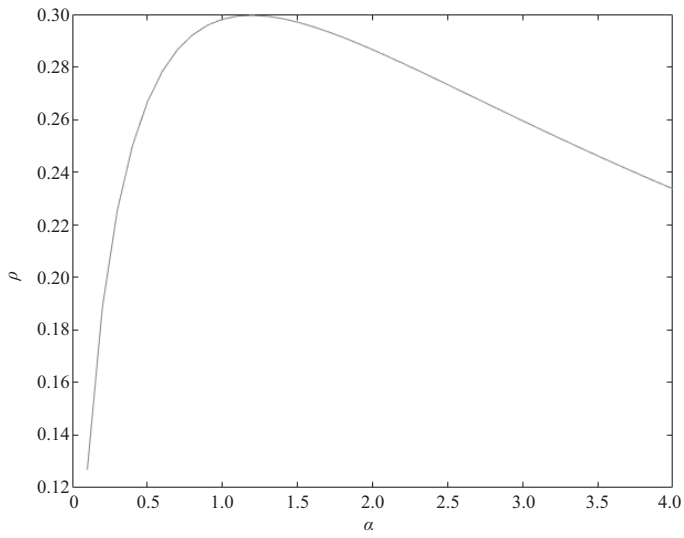

Figure 1 ρ varies over the values of $\alpha > 0$.

Corollary 1. Letting $\alpha = 1.2$, Algorithm 1 yields a subset S satisfying

$$G(S) \geqslant \frac{1}{m} \sum_{i\in[m]} 0.2996 \cdot f_i(S_i^*) - c(S_i^*).$$

Proof. Letting $\frac{\lambda H(\alpha,\lambda)}{H(\alpha,\lambda)+1} = 1$, we get $\lambda = \frac{(4+\alpha+1/\alpha)-\sqrt{(4+\alpha+1/\alpha)^2-4(1+1/\alpha)}}{2}$, where the value of ρ varying over α can be depicted as in Figure 1. It concludes that the minimum value of $\lambda = 3.337$ can be attained by setting $\alpha = 1.2$. Then we get the maximum value $\rho = \frac{1}{\lambda} = 0.2996$. Following the above theorem, we have

$$G(S) \geqslant \frac{1}{m} \sum_{i\in[m]} f_i(T_i^n) - c(T_i^n) \geqslant \frac{1}{m} \sum_{i\in[m]} 0.2996 \cdot f_i(S_i^*) - c(S_i^*).$$

Corollary 2. The memory complexity of Algorithm 1 is $O(\varepsilon^{-1}\log(\ell))$ and the update time per element is at most $O(\varepsilon^{-1}km\log(\ell))$.

5 Conclusion

In this study, we introduce the regularized two-stage submodular maximization problem in streams. We study the two-stage model by adding a regularized term which is used to penalize the chosen elements and intuitively avoid over-fitting. By introducing the distorted add and swap operations, we provide a distorted-replacement-streaming algorithm which obtains a weaker approximation ratio with $\rho = 0.2996$. Consequently, we can further study the regularized model with parameters of generic submodular ratio. Following the proposed framework of the regularized two-stage submodular maximization, we can directly obtain the parameterized result.

Acknowledgements Ruiqi YANG was supported by National Natural Science Foundation of China (Grant No. 12101587), China Postdoctoral Science Foundation (Grant No. 2021M703167), and Fundamental Research Funds for the Central Universities (Grant No. EIE40108X2). Dachuan XU was supported by National Natural Science Foundation of China (Grant No. 12131003) and Beijing Natural Science Foundation Project (Grant No. Z200002). Longkun GUO was supported by National Natural Science Foundation of China (Grant No. 61772005) and Outstanding Youth Innovation Team Project for Universities of Shandong Province (Grant No. 2020KJN008). Dongmei ZHANG was supported by National Natural Science Foundation of China (Grant No. 11871081).

References

1 Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC, 2003. 137–146
2 Lin H, Bilmes J. A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, 2011. 510–520
3 Krause A, McMahan H B, Guestrin C, et al. Robust submodular observation selection. J Machine Learn Res, 2008, 9: 2761–2801

4 Balkanski E, Mirzasoleiman B, Krause A, et al. Learning sparse combinatorial representations via two-stage submodular maximization. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York City, 2016. 2207–2216
5 Badanidiyuru A, Mirzasoleiman B, Karbasi A, et al. Streaming submodular maximization: massive data summarization on the fly. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York City, 2014. 671–680
6 Nemhauser G L, Wolsey L A, Fisher M L. An analysis of approximations for maximizing submodular set functions-I. Math Programming, 1978, 14: 265–294
7 Calinescu G, Chekuri C, Pál M, et al. Maximizing a monotone submodular function subject to a matroid constraint. SIAM J Comput, 2011, 40: 1740–1766
8 Sviridenko M. A note on maximizing a submodular set function subject to a knapsack constraint. Operations Res Lett, 2004, 32: 41–43
9 Buchbinder N, Feldman M, Schwartz R. Online submodular maximization with preemption. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, San Diego, 2015. 1202–1216
10 Norouzi-Fard A, Tarnawski J, Mitrović S, et al. Beyond 1/2-approximation for submodular maximization on massive data streams. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 3826–3835
11 Kazemi E, Mitrovic M, Zadimoghaddam M, et al. Submodular streaming in all its glory: tight approximation, minimum memory and low adaptive complexity. In: Proceedings of International Conference on Machine Learning, Long Beach, 2019. 3311–3320
12 Chekuri C, Gupta S, Quanrud K. Streaming algorithms for submodular function maximization. In: Proceedings of International Colloquium on Automata, Languages and Programming, Kyoto, 2015. 318–330
13 Huang C C, Kakimura N. Improved streaming algorithms for maximizing monotone submodular functions under a knapsack constraint. In: Proceedings of the 16th International Symposium Workshop on Algorithms and Data Structures, Edmonton, 2019. 438–451
14 Huang C C, Kakimura N, Yoshida Y. Streaming algorithms for maximizing monotone submodular functions under a knapsack constraint. Algorithmica, 2020, 82: 1006–1032
15 Kumar R, Moseley B, Vassilvitskii S, et al. Fast greedy algorithms in MapReduce and streaming. ACM Trans Parallel Comput, 2015, 2: 1–22
16 Mirzasoleiman B, Jegelka S, Krause A. Streaming non-monotone submodular maximization: personalized video summarization on the fly. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018. 1379–1386
17 Feldman M. Guess free maximization of submodular and linear sums. In: Proceedings of Workshop on Algorithms and Data Structures, Edmonton, 2019. 380–394
18 Sviridenko M, Vondrák J, Ward J. Optimal approximation for submodular and supermodular optimization with bounded curvature. Math OR, 2017, 42: 1197–1218
19 Harshaw C, Feldman M, Ward J, et al. Submodular maximization beyond non-negativity: guarantees, fast algorithms, and applications. In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019. 2634–2643
20 Kazemi E, Minaee S, Feldman M, et al. Regularized submodular maximization at scale. In: Proceedings of International Conference on Machine Learning, 2021. 5356–5366
21 Stan S, Zadimoghaddam M, Krause A, et al. Probabilistic submodular maximization in sub-linear time. In: Proceedings of International Conference on Machine Learning, Sydney, 2017. 3241–3250
22 Mitrovic M, Kazemi E, Zadimoghaddam M, et al. Data summarization at scale: a two-stage submodular approach. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 3593–3602
23 Gong S, Nong Q, Liu W, et al. Parametric monotone function maximization with matroid constraints. J Glob Optim, 2019, 75: 833–849
24 Yang R, Xu D, Guo L, et al. Parametric streaming two-stage submodular maximization. In: Proceedings of the 16th Annual Conference on Theory and Applications of Models of Computation (TAMC), Changsha, 2020. 193–204