



An Improved Analysis of the Greedy+Singleton Algorithm for k -Submodular Knapsack Maximization

Zhongzheng Tang¹, Jingwen Chen², and Chenhao Wang^{2,3}(✉)

¹ School of Science, Beijing University of Posts and Telecommunications,
Beijing 100876, China

² BNU-HKBU United International College, Zhuhai 519087, China
chenhwang@bnu.edu.cn

³ Beijing Normal University, Zhuhai 519087, China

Abstract. We focus on maximizing a non-negative k -submodular function under a knapsack constraint. As a generalization of submodular functions, a k -submodular function considers k distinct, non-overlapping subsets instead of a single subset as input. We explore the algorithm of Greedy+Singleton, which returns the better one between the best singleton solution and the fully greedy solution. When the function is monotone, we prove that Greedy+Singleton achieves an approximation ratio of $\frac{1}{4}(1 - \frac{1}{e^2}) \approx 0.216$, improving the previous analysis of 0.158 in the literature. Further, we provide the first analysis of Greedy+Singleton for non-monotone functions, and prove an approximation ratio of $\frac{1}{6}(1 - \frac{1}{e^3}) \approx 0.158$.

Keywords: k -submodularity · Greedy · Knapsack · Approximation

1 Introduction

A k -submodular function generalizes a submodular function in a natural way that captures interactions among k subsets. While a submodular function takes a single subset of a finite nonempty set V as input, a k -submodular function considers k disjoint subsets of V , and exhibits the property of diminishing marginal returns common to many problems in operations research.

Given a finite nonempty set V of n items, let $(k+1)^V := \{(X_1, \dots, X_k) \mid X_i \subseteq V \forall i \in [k], X_i \cap X_j = \emptyset \forall i \neq j\}$ be the family of k disjoint sets, where $[k] := \{1, \dots, k\}$. A function $f : (k+1)^V \rightarrow \mathbb{R}$ is k -submodular if and only if for every k -tuples $\mathbf{x} = (X_1, \dots, X_k)$ and $\mathbf{y} = (Y_1, \dots, Y_k)$ in $(k+1)^V$,

$$f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \sqcup \mathbf{y}) + f(\mathbf{x} \sqcap \mathbf{y}),$$

where

$$\mathbf{x} \sqcup \mathbf{y} := (X_1 \cup Y_1 \setminus (\bigcup_{i \neq 1} X_i \cup Y_i), \dots, X_k \cup Y_k \setminus (\bigcup_{i \neq k} X_i \cup Y_i)),$$

$$\mathbf{x} \sqcap \mathbf{y} := (X_1 \cap Y_1, \dots, X_k \cap Y_k).$$

For a k -tuple $\mathbf{x} = (X_1, \dots, X_k) \in (k+1)^V$, we define its size by $|\mathbf{x}| = |\cup_{i \in [k]} X_i|$. We say that $f : (k+1)^V \rightarrow \mathbb{R}$ is *monotone*, if $f(\mathbf{x}) \leq f(\mathbf{y})$ holds for any $\mathbf{x} = (X_1, \dots, X_k)$ and $\mathbf{y} = (Y_1, \dots, Y_k)$ with $X_i \subseteq Y_i$ for $i \in [k]$.

Since Huber and Kolmogorov [6] proposed the notion of k -submodularity one decade ago, there have been increased theoretical and algorithmic interests in the study of k -submodular functions, as various combinatorial optimization problems and practical problems can be formulated as k -submodular function maximization. The applications include influence maximization with k topics in social networks [22], sensor placement with k types of sensors [13], multi-document summarization [11] and multi-class feature selection [27]. For example, given k topics or rumors in a social network, each topic has a different spread model, and we want to select several influential people for each topic to start its spread, in order to maximize the population influenced by at least one topic. This objective function can be modeled as a k -submodular function. More detailed discussions can be found in [22, 27].

As a generalization of the NP-hard submodular maximization problem, the k -submodular maximization problem is also NP-hard. Compared with the submodular maximization where we determine which elements/items are incorporated into the solution, additionally, for k -submodular maximization we need to specify which subsets/dimensions they belongs to. Extensive research has been devoted to developing efficient algorithms and proving their approximation ratios in different settings. In addition to the unconstrained setting [7, 17, 23], researchers also investigate this problem under cardinality constraints [13, 22], matroid constraints [16, 18], and certainly, knapsack constraints [15, 21], which is the focus of this article.

Our Contributions

In this paper, we study the k -submodular maximization problem under a knapsack constraint, called *k -submodular knapsack maximization* (k SKM). Each item $a \in V$ has a cost $c(a)$, and the total cost of the items selected in the solution cannot exceed a given budget $B \in \mathbb{R}_+$. We consider the combination of two natural heuristics for knapsack problems, **Singleton** and **Greedy**. The former returns the best singleton solution $\arg \max_{\mathbf{x}: |\mathbf{x}|=1} f(\mathbf{x})$, that is, it selects a single item and assigns it to a dimension in a way that maximizes the gain of function value. The latter algorithm adds an item to a dimension that maximizes the marginal density (i.e., the marginal gain divided by its cost), until no item fits. Both heuristics are well known to have unbounded approximations, even for linear objectives.

Chen *et al.* [1] first notice that their combination **Greedy+Singleton** achieves a bounded approximation for k SKM, and prove an approximation ratio $\frac{1}{4}(1 - \frac{1}{e}) \approx 0.158$ when the function is monotone. This algorithm compares the outcomes of **Singleton** and **Greedy**, and returns the one with greater value.

We re-consider the **Greedy+Singleton** algorithm and prove an approximation ratio of $\frac{1}{4}(1 - \frac{1}{e^2}) \approx 0.216$ by a more careful analysis for monotone k -submodular functions, improving the result in [1]. Furthermore, for

non-monotone k -submodular functions, we derive an approximation ratio of $\frac{1}{6}(1 - \frac{1}{e^3}) \approx 0.158$.

Though there are several algorithms with proven performance guarantees that are better than ours k SKM in the monotone case, the main advantage of the **Greedy+Singleton** algorithm is the time complexity. Tang *et al.* [21] provide a greedy algorithm with approximation $\frac{1}{2}(1 - \frac{1}{e}) \approx 0.316$ (which combines **Singleton** with a routine that completes all feasible solutions of size 2 greedily), but it takes $O(n^4 k^3)$ queries of the function. Wang and Zhou [22] provide an asymptotically-optimal $(\frac{1}{2} - \epsilon)$ -approximation, but it involves designing a continuous extension of the discrete problem and rounding the fractional solution to recover the discrete solution. Compared with them, **Greedy+Singleton** requires only $O(n^2 k)$ queries.

This paper is organized as follows. In Sect. 2 we present the model and preliminaries. In Sect. 3 we consider the maximization problem without any constraint, of which the result will be used in the analysis of k SKM. In Sect. 4 we analyze the approximation of **Greedy+Singleton** for k SKM. In Sect. 5 we compare it with the method in [1].

Related Work

Huber and Kolmogorov [6] proposed k -submodular functions to express submodularity on choosing k disjoint sets of elements instead of a single set. Recently, this has become a popular subject of research [2, 4, 5, 9, 12, 17].

For the k SKM problem, Tang *et al.* [21] were the first to consider it in the community. When the function is monotone, they provided a $\frac{1}{2}(1 - \frac{1}{e})$ -approximation algorithm that combines **Singleton** with a greedy algorithm that completes all feasible solutions of size 2 greedily. Their analysis framework follows from that of Sviridenko [19] for submodular knapsack maximization problems. Xiao *et al.* [24] later improved the ratio of the same algorithm to $\frac{1}{2}(1 - e^{-2})$ and $\frac{1}{3}(1 - e^{-3})$ for the monotone and non-monotone case, respectively. Wang and Zhou [22] presented an algorithm with asymptotically optimal ratio of $\frac{1}{2} - \epsilon$ by multilinear extension techniques (relaxing the optimization to the continuous space and then rounding the fractional solution). Pham *et al.* [15] proposed a streaming algorithm with approximation ratios $\frac{1}{4} - \epsilon$ and $\frac{1}{5} - \epsilon$ for the monotone and non-monotone cases, respectively, which requires $O(\frac{n}{\epsilon} \log n)$ queries. Other works related to k SKM include [20, 25, 26].

Chen *et al.* first analyzed the performance of **Greedy+Singleton** for k SKM, and proved an approximation ratio $\frac{1}{4}(1 - \frac{1}{e})$. Before them, due to its simplicity and efficiency, **Greedy+Singleton** has received lots of attention for *submodular knapsack maximization*. This algorithm was first suggested in [8] for coverage functions, and adapted to monotone submodular function in [11]. Feldman *et al.* [3] showed that the approximation ratio is within $[0.427, 0.462]$, and Kulik *et al.* [10] presented an improved upper bound of 0.42945. Hence, it limits the approximation ratio of **Greedy+Singleton** for submodular knapsack maximization to the interval $[0.427, 0.42945]$.

The k -submodular maximization problem is also studied in different unconstrained or constrained settings. For unconstrained k -submodular maximization,

Ward and Živný [23] proposed a $\max\{\frac{1}{3}, \frac{1}{1+a}\}$ -approximation algorithm with $a = \max\{1, \sqrt{(k-1)/4}\}$. Later, Iwata *et al.* [7] improved it to $\frac{1}{2}$, which is more recently improved to $\frac{k^2+1}{2k^2+1}$ by Oshima [14]. For monotone k -submodular maximization, Iwata *et al.* [7] also proposed a randomized $\frac{k}{2k-1}$ -approximation algorithm, and showed that the ratio is asymptotically tight. For monotone k -submodular maximization under a total size constraint (i.e., $\sum_{i \in [k]} |X_i| \leq B$ for an integer budget B), Ohsaka and Yoshida [13] proposed a $\frac{1}{2}$ -approximation algorithm, and a $\frac{1}{3}$ -approximation algorithm for that under individual size constraints (i.e., $|X_i| \leq B_i \forall i \in [k]$ with budgets B_i). Under a matroid constraint, Sakaue [16] proposed a $\frac{1}{2}$ -approximation algorithm for the monotone case, which is asymptotically tight, and Sun *et al.* [18] gave a $\frac{1}{3}$ -approximation algorithm for the non-monotone case.

2 Preliminaries

We introduce more characteristics of k -submodular functions. If two k -tuples $\mathbf{x} = (X_1, \dots, X_k)$ and $\mathbf{y} = (Y_1, \dots, Y_k)$ in $(k+1)^V$ with $X_i \subseteq Y_i$ for each $i \in [k]$, we denote $\mathbf{x} \preceq \mathbf{y}$. Define the marginal gain when adding item a to the i -th dimension of $\mathbf{x} = (X_1, \dots, X_k)$ to be

$$\Delta_{a,i}(\mathbf{x}) := f(X_1, \dots, X_{i-1}, X_i \cup \{a\}, X_{i+1}, \dots, X_k) - f(\mathbf{x}),$$

and thus $\frac{\Delta_{a,i}(\mathbf{x})}{c(a)}$ is the marginal density. A k -submodular function f clearly satisfies the *orthant submodularity*

$$\Delta_{a,i}f(\mathbf{x}) \geq \Delta_{a,i}f(\mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in (k+1)^V \text{ with } \mathbf{x} \preceq \mathbf{y}, a \notin \bigcup_{j \in [k]} Y_j, i \in [k],$$

and the *pairwise monotonicity*

$$\Delta_{a,i_1}f(\mathbf{x}) + \Delta_{a,i_2}f(\mathbf{x}) \geq 0, \forall \mathbf{x} \in (k+1)^V \text{ with } a \notin \bigcup_{j \in [k]} X_j, i_1, i_2 \in [k], i_1 \neq i_2.$$

Ward and Živný [23] show that the converse is also true.

Lemma 1 ([23]). *A function $f : (k+1)^V \rightarrow \mathbb{R}$ is k -submodular if and only if f is orthant submodular and pairwise monotone.*

It is easy to see that when f is monotone, the k -submodularity degenerates into orthant submodularity.

Every k -tuple $\mathbf{x} = (X_1, \dots, X_k) \in (k+1)^V$ uniquely corresponds to a set $S = \{(a, d) \mid \exists d \in [k] a \in X_d\}$ that consists of *item-index pairs*. That is, an item-index pair (a, d) belongs to S (called a *solution*) if and only if there is an index d so that $a \in X_d$. From now on, with a slight abuse of notations, we write \mathbf{x} and its corresponding solution S interchangeably, for example, $\Delta_{a,d}(S)$ means the marginal gain $f(S \cup \{(a, d)\}) - f(S)$. For any solution S , we define

Algorithm 1. Unconstrained Greedy**Input:** Set V' , k -submodular function f **Output:** A solution $S \in (k+1)^V$

- 1: $S \leftarrow \emptyset$
- 2: **for** each item $a \in V'$ **do**
- 3: $d_a \leftarrow \arg \max_{d \in [k]} \Delta_{a,d}(S)$
- 4: $S \leftarrow S \cup \{(a, d_a)\}$
- 5: **end for**
- 6: **return** S

$U(S) := \{a \in V \mid \exists d \in [k] (a, d) \in S\}$ to be the set of items included, and the *size* of S is $|S| = |U(S)|$. In this paper, let f be a non-negative k -submodular function, and we further assume w.l.o.g. that $f(\emptyset) = 0$.

We point out the following lemma that will repeatedly and implicitly used in our analysis.

Lemma 2 ([21]). *For any solutions S, S' with $S \subseteq S'$, we have*

$$f(S') - f(S) \leq \sum_{(a,d) \in S' \setminus S} \Delta_{a,d}(S).$$

3 A Key Lemma for Unconstrained k -Submodular Maximization

In this section, we consider the problem of maximizing the function value in the unconstrained setting, for an arbitrary subset of items $V' = \{e_1, e_2, \dots, e_m\} \subseteq V$. Algorithm 1 (**Unconstrained Greedy**) considers items in V' in an arbitrary order, and assigns each item the *best* index that brings the largest marginal gain in each iteration. We will introduce a lemma that is important for the analysis in Sect. 4 for k SKM.

Let $T = \{(e_1, d_1^*), \dots, (e_m, d_m^*)\}$ be an optimal solution that maximizes the function value over V' (such an optimal solution must exist due to the pairwise monotonicity). We dictate that **Unconstrained Greedy** considers the items in an order of e_1, e_2, \dots, e_m , and denote the returned greedy solution by $S = \{(e_1, d_1), \dots, (e_m, d_m)\}$.

For $j = 0, 1, \dots, m$, define

$$S_j = \{(e_1, d_1), \dots, (e_j, d_j)\} \text{ and} \quad (1)$$

$$T_j = (T \setminus \{(e_1, d_1^*), \dots, (e_j, d_j^*)\}) \cup S_j. \quad (2)$$

That is, S_j is the first j item-index pairs in the greedy solution S , and T_j is obtained from the optimal solution T by replacing the first j item-index pairs with S_j . Clearly, $S_0 = \emptyset, S_m = S, T_0 = T$ and $T_m = S$.

The following key lemma bounds the optimal value $f(T)$ in terms of $f(S_t)$ and marginal gains. This conclusion is firstly noticed by Ward and Živný (implicitly in Theorem 5.1 [23]) and formalized by Xiao *et al.* [24]. For completeness, we write down the proof and credit [23, 24].

We point out the following lemma

Lemma 3. For $t = 0, 1, \dots, m$,

- (a) if f is monotone, then $f(T) \leq 2f(S_t) + \sum_{(a,d) \in T_t \setminus S_t} \Delta_{a,d}(S_t)$;
(b) if f is non-monotone, then $f(T) \leq 3f(S_t) + \sum_{(a,d) \in T_t \setminus S_t} \Delta_{a,d}(S_t)$;

Proof. For $j = 0, \dots, t-1$, we introduce an intermediate $P_j := T_j \setminus (e_{j+1}, d_{j+1}^*) = T_{j+1} \setminus (e_{j+1}, d_{j+1})$. That is, P_j consists of $m-1$ items (excluding e_{j+1}), where the indices of items e_1, \dots, e_j coincide those in S , and the indices of other items coincide those in T . Then

$$f(T_j) = f(P_j) + \Delta_{e_{j+1}, d_{j+1}^*}(P_j),$$

$$f(T_{j+1}) = f(P_j) + \Delta_{e_{j+1}, d_{j+1}}(P_j).$$

When f is monotone, the difference of $f(T_j)$ and $f(T_{j+1})$ is

$$\begin{aligned} f(T_j) - f(T_{j+1}) &= \Delta_{e_{j+1}, d_{j+1}^*}(P_j) - \Delta_{e_{j+1}, d_{j+1}}(P_j) \\ &\leq \Delta_{e_{j+1}, d_{j+1}^*}(S_j) \end{aligned} \tag{3}$$

$$\leq \Delta_{e_{j+1}, d_{j+1}}(S_j) \tag{4}$$

$$= f(S_{j+1}) - f(S_j).$$

Equation (3) follows from the fact of $S_j \subseteq P_j$ and the monotonicity of f . Equation (4) follows from the fact that the greedy algorithm always assign the index with maximum marginal gain to the item considered, and (e_{j+1}, d_{j+1}) is the $(j+1)$ -th pair added. Summing this inequality from $j = 0$ to $t-1$, we obtain

$$f(T_0) - f(T_t) \leq f(S_t) - f(S_0) = f(S_t).$$

Since $S_t \subseteq T_t$ and Lemma 2, we have

$$f(T) \leq f(S_t) + f(T_t) \leq 2f(S_t) + \sum_{(a,d) \in T_t \setminus S_t} \Delta_{a,d}(S_t).$$

When f is non-monotone, Eq. (3) no longer holds. Instead, we bound the difference of $f(T_j)$ and $f(T_{j+1})$ by

$$\begin{aligned} f(T_j) - f(T_{j+1}) &= \Delta_{e_{j+1}, d_{j+1}^*}(P_j) - \Delta_{e_{j+1}, d_{j+1}}(P_j) \\ &= 2\Delta_{e_{j+1}, d_{j+1}^*}(P_j) - [\Delta_{e_{j+1}, d_{j+1}^*}(P_j) + \Delta_{e_{j+1}, d_{j+1}}(P_j)] \\ &\leq 2\Delta_{e_{j+1}, d_{j+1}^*}(P_j) \\ &\leq 2\Delta_{e_{j+1}, d_{j+1}^*}(S_j) \leq 2\Delta_{e_{j+1}, d_{j+1}}(S_j) \\ &= 2f(S_{j+1}) - 2f(S_j), \end{aligned} \tag{5}$$

where Eq. (5) follows from the pairwise monotonicity. Summing it from $j = 0$ to $t - 1$, we obtain

$$f(T_0) - f(T_t) \leq 2f(S_t) - 2f(S_0) = 2f(S_t).$$

Since $S_t \subseteq T_t$ and Lemma 2, we have

$$f(T) \leq 2f(S_t) + f(T_t) \leq 3f(S_t) + \sum_{(a,d) \in T_t \setminus S_t} \Delta_{a,d}(S_t).$$

□

Letting $t = m$ in the above lemma, it is easy to see that the greedy solution S is 2-approximation of $f(T)$ when f is monotone, and 3-approximation of $f(T)$ when f is non-monotone.

4 Greedy+Singleton for k -Submodular Knapsack

We consider the k SKM problem. Each item $a \in V$ has a cost $c(a)$, and the total cost of selected items must not exceed a given budget $B \in \mathbb{R}_+$. For any solution $S \in (k+1)^V$, define $c(S) = \sum_{a \in U(S)} c(a)$ to be the total cost of all items in S .

We consider **Greedy+Singleton** (Algorithm 2). It returns the better solution between **Greedy** and **Singleton**, where the former greedily chooses the item-index pair of maximum marginal density in every iteration until no item fits (Line 2–11), and the latter chooses the single item-index pair of maximum marginal gain (Line 1).

Next, we prove approximation ratios $\frac{1}{4}(1 - \frac{1}{e^2})$ and $\frac{1}{6}(1 - \frac{1}{e^3})$ for the monotone and non-monotone cases, respectively. The general framework follows from Khuller *et al.* [8] for the budgeted maximum coverage problem, which gives a $\frac{1}{2}(1 - \frac{1}{e})$ approximation for the submodular knapsack maximization. We adapt it to k SKM, and utilize the characteristics of k -submodularity.

Algorithm 2. Greedy+Singleton

- 1: Let $S^* \in \arg \max_{S: |S|=1, c(S) \leq B} f(S)$ be a singleton solution giving the largest value.
 - 2: $G_0 \leftarrow \emptyset, V^0 \leftarrow V$
 - 3: **for** t from 1 to n **do**
 - 4: Let $(a_t, d_t) = \arg \max_{a \in V^{t-1}, d \in [k]} \frac{\Delta_{a,d}(G_{t-1})}{c(a)}$ be the pair maximizing the marginal density
 - 5: **if** $c(G_{t-1}) + c(a_t) \leq B$ **then**
 - 6: $G_t = G_{t-1} \cup \{(a_t, d_t)\}$
 - 7: **else**
 - 8: $G_t = G_{t-1}$
 - 9: **end if**
 - 10: $V^t = V^{t-1} \setminus \{a_t\}$
 - 11: **end for**
 - 12: $S^* \leftarrow G_n$ if $f(G_n) > f(S^*)$
 - 13: **return** S^*
-

Let OPT be the optimal solution, and $f(OPT)$ be the optimal value. In each iteration $t = 1, \dots, n$, a pair (a_t, d_t) is considered, and G_t is called the partial greedy solution. Let $l + 1$ be the first time when Algorithm 2 does not add an item in $U(OPT)$ to the current solution because its addition would violate the budget (i.e., $a_{l+1} \in U(OPT)$ and $c(a_{l+1}) + c(G_l) > B$). We can further assume that $l + 1$ is the first time t for which $G_t = G_{t-1}$. This assumption is without loss of generality, because if it happens earlier for some $t' < l + 1$, then $a_{t'}$ does not belong to the optimal solution T , nor the approximate solution we are interested in; thus, we can remove $a_{t'}$ from the ground set V , without affecting the analysis, the optimal solution T , and the approximate solution. Thus, we have $G_t = G_{t-1} \cup \{(a_t, d_t)\}$ for $t = 1, \dots, l$.

For each $t = 1, \dots, l$, we define $\tilde{G}_t = G_t$ to be the partial greedy solution after the t -th iteration, and define $\tilde{G}_{l+1} = G_l \cup \{(a_{l+1}, d_{l+1})\}$ to be the solution obtained by adding (a_{l+1}, d_{l+1}) to G_l . Note that \tilde{G}_{l+1} violates the budget, and $G_l = G_{l+1} \neq \tilde{G}_{l+1}$ by our assumption.

Next, we prove the approximation ratio of **Greedy+Singleton** by a series of lemmas and Theorem 1. In Lemma 4, we show that a selected item which occupies a large proportion of the budget gives a good approximation. In Lemma 5 we bound the marginal gain in every iteration, and then Lemma 6 gives a lower bound on every $f(\tilde{G}_t)$.

Lemma 4. *For $t = 1, \dots, l$, if $c(a_t) \geq \alpha \cdot B$, then the partial greedy solution \tilde{G}_t is $\min\{\frac{1}{2}, \alpha\}$ -approximation if f is monotone, and $\min\{\frac{1}{3}, \alpha\}$ -approximation if f is non-monotone.*

Proof. For each $t = 1, \dots, l$, we consider the unconstrained maximization over the items in $V' := U(\tilde{G}_{t-1}) \cup U(OPT) = \{e_1, \dots, e_m\}$. Assume w.l.o.g. that $e_1 = a_1, e_2 = a_2, \dots, e_{t-1} = a_{t-1}$. Let Algorithm 1 consider the items in the order of e_1, \dots, e_m . Recall the notations in Eq. (1) and (2), and note that $\tilde{G}_j = S_j$ for $j = 1, \dots, t-1$, that is, the partial greedy solutions in Algorithm 2 coincide those in Algorithm 1. Denote by OPT' the optimal solution of the unconstrained maximization over $U(\tilde{G}_{t-1}) \cup U(OPT)$, and we apply Lemma 3 to bound $f(OPT')$.

When f is monotone, by Lemma 3 we have

$$\begin{aligned} f(OPT) \leq f(OPT') &\leq 2f(\tilde{G}_{t-1}) + \sum_{(a,d) \in T_{t-1} \setminus \tilde{G}_{t-1}} \Delta_{a,d}(\tilde{G}_{t-1}) \\ &\leq 2f(\tilde{G}_{t-1}) + \sum_{(a,d) \in T_{t-1} \setminus \tilde{G}_{t-1}} c(a) \cdot \frac{\Delta_{a,d}(\tilde{G}_{t-1})}{c(a_t)} \end{aligned} \quad (6)$$

$$\leq 2f(\tilde{G}_{t-1}) + \frac{\Delta_{a_t, d_t}(\tilde{G}_{t-1})}{c(a_t)} \cdot B, \quad (7)$$

where Eq. (6) is because (a_t, d_t) is the pair of maximum marginal density by the greedy algorithm, and Eq. (7) is because the items in $T_{t-1} \setminus \tilde{G}_{t-1}$ must belong to OPT and their total cost is at most B . Combining with the value of the partial

greedy solution $f(\bar{G}_t) = f(\bar{G}_{t-1}) + \Delta_{a_t, d_t}(\bar{G}_{t-1})$, it is easy to see that

$$\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{2} \cdot \frac{f(\bar{G}_{t-1}) + \Delta_{a_t, d_t}(\bar{G}_{t-1})}{f(\bar{G}_{t-1}) + \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})B}{2c(a_t)}}.$$

If $\frac{B}{2c(a_t)} \leq 1$, then clearly $\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{2}$. If $\frac{B}{2c(a_t)} > 1$, then

$$\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{2} \cdot \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})}{\frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})B}{2c(a_t)}} = \frac{c(a_t)}{B} \geq \alpha.$$

Therefore, \bar{G}_t is $\min\{\frac{1}{2}, \alpha\}$ -approximation.

When f is non-monotone, using Lemma 3 for $t = 1, \dots, l$, similarly we have

$$\begin{aligned} f(OPT) &\leq f(OPT') \leq 3f(\bar{G}_{t-1}) + \sum_{(a,d) \in T_{t-1} \setminus \bar{G}_{t-1}} \Delta_{a,d}(\bar{G}_{t-1}) \\ &\leq 3f(\bar{G}_{t-1}) + \sum_{(a,d) \in T_{t-1} \setminus \bar{G}_{t-1}} c(a) \cdot \frac{\Delta_{a,d}(\bar{G}_{t-1})}{c(a_t)} \\ &\leq 3f(\bar{G}_{t-1}) + \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})}{c(a_t)} \cdot B, \end{aligned}$$

Combining with the value of the partial greedy solution $f(\bar{G}_t) = f(\bar{G}_{t-1}) + \Delta_{a_t, d_t}(\bar{G}_{t-1})$, it is easy to see that

$$\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{3} \cdot \frac{f(\bar{G}_{t-1}) + \Delta_{a_t, d_t}(\bar{G}_{t-1})}{f(\bar{G}_{t-1}) + \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})B}{3c(a_t)}}.$$

If $\frac{B}{3c(a_t)} \leq 1$, then clearly $\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{3}$. If $\frac{B}{3c(a_t)} > 1$, then

$$\frac{f(\bar{G}_t)}{f(OPT)} \geq \frac{1}{3} \cdot \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})}{\frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})B}{3c(a_t)}} = \frac{c(a_t)}{B} \geq \alpha.$$

Therefore, \bar{G}_t is $\min\{\frac{1}{3}, \alpha\}$ -approximation. \square

The following lemma bounds the marginal gain in every iteration.

Lemma 5. For each $t = 1, \dots, l+1$,

(a) if f is monotone, then

$$f(\bar{G}_t) - f(\bar{G}_{t-1}) \geq \frac{c(a_t)}{B} (f(OPT) - 2f(\bar{G}_{t-1}))$$

(b) if f is non-monotone, then

$$f(\bar{G}_t) - f(\bar{G}_{t-1}) \geq \frac{c(a_t)}{B} (f(OPT) - 3f(\bar{G}_{t-1}))$$

Proof. As in the proof of Lemma 4, we again consider the unconstrained maximization over $U(\bar{G}_{t-1}) \cup U(OPT)$ for each $t = 1, \dots, l+1$, and assume that the partial greedy solutions in Algorithm 2 coincide those in Algorithm 1. Denote by OPT' the optimal solution of this unconstrained maximization problem.

When f is monotone, by Lemma 3 (a), for $t = 1, \dots, l+1$ we have

$$\begin{aligned} f(OPT) &\leq f(OPT') \leq 2 \cdot f(\bar{G}_{t-1}) + \sum_{(a,d) \in T_{t-1} \setminus \bar{G}_{t-1}} \Delta_{a,d}(\bar{G}_{t-1}) \\ &\leq 2 \cdot f(\bar{G}_{t-1}) + B \cdot \frac{\Delta_{a_t, d_t}(\bar{G}_{t-1})}{c(a_t)} \\ &= 2 \cdot f(\bar{G}_{t-1}) + B \cdot \frac{f(\bar{G}_t) - f(\bar{G}_{t-1})}{c(a_t)}, \end{aligned}$$

where the last inequality follows from the facts that the marginal density is maximized in each iteration and the capacity remained is at most B . Then immediately we have $f(\bar{G}_t) - f(\bar{G}_{t-1}) \geq \frac{c(a_t)}{B} (f(OPT) - 2f(\bar{G}_{t-1}))$.

When f is non-monotone, by Lemma 3 (b), a similar analysis gives

$$f(OPT) \leq 3 \cdot f(\bar{G}_{t-1}) + B \cdot \frac{f(\bar{G}_t) - f(\bar{G}_{t-1})}{c(a_t)}.$$

□

Lemma 6. *For each $t = 1, \dots, l+1$, we have*

$$f(\bar{G}_t) \geq (1 - x_t) \cdot f(OPT),$$

where $x_1 = 1 - \frac{c(a_1)}{B}$, $x_t = (1 - \frac{2c(a_t)}{B})x_{t-1} + \frac{c(a_t)}{B}$ if f is monotone, and $x_t = (1 - \frac{3c(a_t)}{B})x_{t-1} + \frac{2c(a_t)}{B}$ if f is non-monotone.

Proof. We prove it by induction. Firstly, when $t = 1$, clearly we have $f(\bar{G}_1) \geq \frac{c(a_1)}{B} f(OPT)$. Assume that the statement holds for iterations $1, 2, \dots, t-1$. We show that it also holds for iteration t . When f is monotone, by Lemma 5 (a),

$$\begin{aligned} f(\bar{G}_t) &= f(\bar{G}_{t-1}) + f(\bar{G}_t) - f(\bar{G}_{t-1}) \\ &\geq f(\bar{G}_{t-1}) + \frac{c(a_t)}{B} (f(OPT) - 2f(\bar{G}_{t-1})) \\ &= (1 - \frac{2c(a_t)}{B})f(\bar{G}_{t-1}) + \frac{c(a_t)}{B} f(OPT) \\ &\geq (1 - \frac{2c(a_t)}{B})(1 - x_{t-1}) \cdot f(OPT) + \frac{c(a_t)}{B} f(OPT) \\ &= [1 - ((1 - \frac{2c(a_t)}{B})x_{t-1} + \frac{c(a_t)}{B})] f(OPT). \end{aligned}$$

When f is non-monotone, a similar analysis follows from Lemma 5 (b). □

It is not hard to see that the recurrence relation $x_t = (1 - \frac{2c(a_t)}{B})x_{t-1} + \frac{c(a_t)}{B}$ with initial state $x_1 = 1 - \frac{c(a_1)}{B}$ can be written as

$$x_t - \frac{1}{2} = (1 - \frac{2c(a_t)}{B})x_{t-1} - \frac{1}{2}(1 - \frac{2c(a_t)}{B}) = (1 - \frac{2c(a_t)}{B})(x_{t-1} - \frac{1}{2}).$$

Hence, for the monotone case we can easily get a general formula

$$x_t = (\frac{1}{2} - \frac{c(a_1)}{B}) \prod_{j=2}^t (1 - \frac{2c(a_j)}{B}) + \frac{1}{2}. \quad (8)$$

For the non-monotone case, similarly we can write the recurrence relation as

$$x_t - \frac{2}{3} = (1 - \frac{3c(a_t)}{B})x_{t-1} - \frac{2}{3}(1 - \frac{3c(a_t)}{B}) = (1 - \frac{3c(a_t)}{B})(x_{t-1} - \frac{2}{3}),$$

and get a general formula

$$x_t = (\frac{1}{3} - \frac{c(a_1)}{B}) \prod_{j=2}^t (1 - \frac{3c(a_j)}{B}) + \frac{2}{3}. \quad (9)$$

Now we are ready to prove our main theorem.

Theorem 1. *For the k SKM, **Greedy+Singleton** achieves an approximation ratio of $\frac{1}{4}(1 - \frac{1}{e^2}) \approx 0.216$ and $\frac{1}{6}(1 - \frac{1}{e^3}) \approx 0.158$ when the function is monotone and non-monotone, respectively, within $O(n^2k)$ queries.*

Proof. When f is monotone, by Lemma 6 and Eq. (8), we have

$$\begin{aligned} f(\bar{G}_{l+1}) &\geq (1 - x_{l+1}) \cdot f(OPT) \\ &= \left(\frac{1}{2} - \left(\frac{1}{2} - \frac{c(a_1)}{B} \right) \prod_{j=2}^{l+1} \left(1 - \frac{2c(a_j)}{B} \right) \right) \cdot f(OPT) \\ &= \left(\frac{1}{2} - \frac{1}{2} \prod_{j=1}^{l+1} \left(1 - \frac{2c(a_j)}{B} \right) \right) \cdot f(OPT). \end{aligned} \quad (10)$$

If $1 - \frac{2c(a_j)}{B} \geq 0$ for all $j \in [l+1]$, since $c(\bar{G}_{l+1}) = c(\bar{G}_l) + c(a_{l+1}) > B$, we obtain

$$\begin{aligned} f(\bar{G}_{l+1}) &\geq \left(\frac{1}{2} - \frac{1}{2} \prod_{j=1}^{l+1} \left(1 - \frac{2c(a_j)}{c(\bar{G}_{l+1})} \right) \right) \cdot f(OPT) \\ &\geq \left(\frac{1}{2} - \frac{1}{2} \cdot \left(1 - \frac{2}{l+1} \right)^{l+1} \right) \cdot f(OPT) \\ &\geq \left(\frac{1}{2} - \frac{1}{2e^2} \right) \cdot f(OPT). \end{aligned} \quad (11)$$

If $1 - \frac{2c(a_j)}{B} < 0$ for exactly one $j \in [l+1]$ and $1 - \frac{2c(a_j)}{B} \geq 0$ for all $i \neq j$, it immediately follows from Eq. (10) that $f(\bar{G}_{l+1}) \geq \frac{1}{2}f(OPT)$. It remains to

consider the case when $1 - \frac{2c(a_j)}{B} < 0$ for exactly one $j \in [l]$ and $1 - \frac{2c(a_{l+1})}{B} < 0$. By Lemma 4, the large cost of item a_j implies that \bar{G}_j has an approximation at least $\min\{\frac{1}{2}, \frac{c(a_j)}{B}\} \geq \frac{1}{2}$. By the monotonicity we have $f(\bar{G}_{l+1}) \geq f(\bar{G}_j) \geq \frac{1}{2}f(OPT)$.

Hence, we always have

$$f(\bar{G}_{l+1}) = f(\bar{G}_l) + \Delta_{a_{l+1}, d_{l+1}}(\bar{G}_l) \geq \left(\frac{1}{2} - \frac{1}{2e^2}\right) \cdot f(OPT).$$

Note that $\Delta_{a_{l+1}, d_{l+1}}(\bar{G}_l)$ is no more than the maximum profit of a single item, i.e., the outcome of **Singleton**, say (a^*, d^*) . Therefore, the better solution between \bar{G}_l and $\{(a^*, d^*)\}$ has a value

$$\max\{f(\bar{G}_l), f(\{(a^*, d^*)\})\} \geq \frac{1}{2} \left(\frac{1}{2} - \frac{1}{2e^2}\right) \cdot f(OPT).$$

Since \bar{G}_l is a part of the solution returned by **Greedy+Singleton** when **Greedy** performs better than **Singleton**, it establishes an approximation ratio $\frac{1}{4}(1 - \frac{1}{e^2})$.

When f is non-monotone, by Lemma 6 and Eq. (9), we have

$$\begin{aligned} f(\bar{G}_{l+1}) &\geq (1 - x_{l+1}) \cdot f(OPT) \\ &= \left(\frac{1}{3} - \left(\frac{1}{3} - \frac{c(a_1)}{B}\right) \prod_{j=2}^{l+1} \left(1 - \frac{3c(a_j)}{B}\right)\right) \cdot f(OPT) \\ &= \left(\frac{1}{3} - \frac{1}{3} \prod_{j=1}^{l+1} \left(1 - \frac{3c(a_j)}{B}\right)\right) \cdot f(OPT). \end{aligned} \quad (12)$$

If $1 - \frac{3c(a_j)}{B} \geq 0$ for all $j \in [l+1]$, since $c(\bar{G}_{l+1}) > B$, we obtain

$$\begin{aligned} f(\bar{G}_{l+1}) &\geq \left(\frac{1}{3} - \frac{1}{3} \prod_{j=1}^{l+1} \left(1 - \frac{3c(a_j)}{c(\bar{G}_{l+1})}\right)\right) \cdot f(OPT) \\ &\geq \left(\frac{1}{3} - \frac{1}{3} \cdot \left(1 - \frac{3}{l+1}\right)^{l+1}\right) \cdot f(OPT) \\ &\geq \left(\frac{1}{3} - \frac{1}{3e^3}\right) \cdot f(OPT). \end{aligned}$$

If $1 - \frac{3c(a_j)}{B} < 0$ holds for one j or three j 's in $[l+1]$, then it immediately follows from Eq. (12) that $f(\bar{G}_{l+1}) \geq \frac{1}{3}f(OPT)$. It remains to consider the case when $1 - \frac{3c(a_{j_1})}{B} < 0$, $1 - \frac{3c(a_{j_2})}{B} < 0$, and $1 - \frac{3c(a_i)}{B} \geq 0$ for all $i \notin \{j_1, j_2\}$. Assume $j_1 \in [l]$. By Lemma 4, the large cost of item a_{j_1} implies that \bar{G}_{j_1} has an approximation at least $\min\{\frac{1}{3}, \frac{c(a_{j_1})}{B}\} \geq \frac{1}{3}$. By the pairwise monotonicity and the greedy procedure we know that the function value of partial greedy solutions is non-decreasing, and thus $f(\bar{G}_{l+1}) \geq f(\bar{G}_{j_1}) \geq \frac{1}{3}f(OPT)$.

Therefore, **Greedy+Singleton** has a value at least

$$\max\{f(\bar{G}_l), f(\{(a^*, i^*)\})\} \geq \frac{1}{2} \left(\frac{1}{3} - \frac{1}{3e^3}\right) \cdot f(OPT).$$

□

5 Conclusion

We provided a novel analysis of **Greedy+Singleton** for the k SKM, and proved approximation ratios $\frac{1}{4}(1 - \frac{1}{e^2})$ and $\frac{1}{6}(1 - \frac{1}{e^3})$ for monotone and non-monotone functions, respectively. Compared with the $\frac{1}{4}(1 - \frac{1}{e})$ -approximation in [1], our improvement heavily relies on the key proposition of Lemma 3, which gives upper bounds on the optimum $f(T)$ (for unconstrained maximization) in terms of every partial greedy solution S_t , instead of the simple 2-approximation achieved by the final greedy solution in [1]. Moreover, our Lemma 4 shows that a selected item with a large cost gives a good approximation, which is also useful for proving the improved approximation ratios. Future directions include further improving the approximation ratio of **Greedy+Singleton** and looking for other efficient algorithms.

Acknowledgements. This work is partially supported by Artificial Intelligence and Data Science Research Hub, BNU-HKBU United International College (UIC), No. 2020KSYS007, and by a grant from UIC (No. UICR0400025-21). Zhongzheng Tang is supported by National Natural Science Foundation of China under Grant No. 12101069. Chenhao Wang is supported by NSFC under Grant No. 12201049, and is also supported by UIC grants of UICR0400014-22, UICR0200008-23 and UICR0700036-22.

References

1. Chen, J., Tang, Z., Wang, C.: Monotone k -submodular knapsack maximization: an analysis of the Greedy+Singleton algorithm. In: Ni, Q., Wu, W. (eds.) AAIM 2022. LNCS, vol. 13513, pp. 144–155. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16081-3_13
2. Ene, A., Nguyen, H.: Streaming algorithm for monotone k -submodular maximization with cardinality constraints. In: Proceedings of the 39th International Conference on Machine Learning (ICML), pp. 5944–5967. PMLR (2022)
3. Feldman, M., Nutov, Z., Shoham, E.: Practical budgeted submodular maximization. *Algorithmica* 1–40 (2022)
4. Gridchyn, I., Kolmogorov, V.: Potts model, parametric maxflow and k -submodular functions. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2320–2327 (2013)
5. Hirai, H., Iwamasa, Y.: On k -submodular relaxation. *SIAM J. Discret. Math.* **30**(3), 1726–1736 (2016)
6. Huber, A., Kolmogorov, V.: Towards minimizing k -submodular functions. In: Mahjoub, A.R., Markakis, V., Milis, I., Paschos, V.T. (eds.) ISCO 2012. LNCS, vol. 7422, pp. 451–462. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32147-4_40
7. Iwata, S., Tanigawa, S., Yoshida, Y.: Improved approximation algorithms for k -submodular function maximization. In: Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 404–413 (2016)
8. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. *Inf. Process. Lett.* **70**(1), 39–45 (1999)
9. Kudla, J., Živný, S.: Sparsification of monotone k -submodular functions of low curvature. arXiv preprint [arXiv:2302.03143](https://arxiv.org/abs/2302.03143) (2023)

10. Kulik, A., Schwartz, R., Shachnai, H.: A refined analysis of submodular greedy. *Oper. Res. Lett.* **49**(4), 507–514 (2021)
11. Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 912–920 (2010)
12. Nguyen, L., Thai, M.T.: Streaming k -submodular maximization under noise subject to size constraint. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 7338–7347. PMLR (2020)
13. Ohsaka, N., Yoshida, Y.: Monotone k -submodular function maximization with size constraints. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 1, pp. 694–702 (2015)
14. Oshima, H.: Improved randomized algorithm for k -submodular function maximization. *SIAM J. Discret. Math.* **35**(1), 1–22 (2021)
15. Pham, C.V., Vu, Q.C., Ha, D.K., Nguyen, T.T., Le, N.D.: Maximizing k -submodular functions under budget constraint: applications and streaming algorithms. *J. Comb. Optim.* **44**(1), 723–751 (2022)
16. Sakaue, S.: On maximizing a monotone k -submodular function subject to a matroid constraint. *Discret. Optim.* **23**, 105–113 (2017)
17. Soma, T.: No-regret algorithms for online k -submodular maximization. In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1205–1214. PMLR (2019)
18. Sun, Y., Liu, Y., Li, M.: Maximization of k -submodular function with a matroid constraint. In: Du, D.Z., Du, D., Wu, C., Xu, D. (eds.) *TAMC 2022*. LNCS, vol. 13571, pp. 1–10. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20350-3_1
19. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.* **32**(1), 41–43 (2004)
20. Tang, Z., Wang, C., Chan, H.: Monotone k -submodular secretary problems: cardinality and knapsack constraints. *Theor. Comput. Sci.* **921**, 86–99 (2022)
21. Tang, Z., Wang, C., Chan, H.: On maximizing a monotone k -submodular function under a knapsack constraint. *Oper. Res. Lett.* **50**(1), 28–31 (2022)
22. Wang, B., Zhou, H.: Multilinear extension of k -submodular functions. arXiv preprint [arXiv:2107.07103](https://arxiv.org/abs/2107.07103) (2021)
23. Ward, J., Živný, S.: Maximizing k -submodular functions and beyond. *ACM Trans. Algorithms* **12**(4), 1–26 (2016)
24. Xiao, H., Liu, Q., Zhou, Y., Li, M.: Small notes on k -submodular maximization with a knapsack constraint. Technical report (2023)
25. Yu, K., Li, M., Zhou, Y., Liu, Q.: Guarantees for maximization of k -submodular functions with a knapsack and a matroid constraint. In: Ni, Q., Wu, W. (eds.) *AAIM 2022*. LNCS, vol. 13513, pp. 156–167. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16081-3_14
26. Yu, K., Li, M., Zhou, Y., Liu, Q.: On maximizing monotone or non-monotone k -submodular functions with the intersection of knapsack and matroid constraints. *J. Comb. Optim.* **45**(3), 1–21 (2023)
27. Yu, Q., Küçükyavuz, S.: An exact cutting plane method for k -submodular function maximization. *Discret. Optim.* **42**, 100670 (2021)