Sequence Selection by Pareto Optimization

Chao Qian[1], **Chao Feng**[1], **Ke Tang**[2]

[1] Anhui Province Key Lab of Big Data Analysis and Application,
University of Science and Technology of China, Hefei 230027, China

[2] Shenzhen Key Lab of Computational Intelligence, Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China

chaoqian@ustc.edu.cn, chaofeng@mail.ustc.edu.cn, tangk3@sustc.edu.cn

Abstract

The problem of selecting a sequence of items from a universe that maximizes some given objective function arises in many real-world applications. In this paper, we propose an anytime randomized iterative approach POSEQSEL, which maximizes the given objective function and minimizes the sequence length simultaneously. We prove that for any previously studied objective function, POSE-QSEL using a reasonable time can always reach or improve the best known approximation guarantee. Empirical results exhibit the superior performance of POSEQSEL.

1 Introduction

The subset selection problem, which selects a subset of size at most k from a total set of n items for maximizing some given objective function f, arises in many applications, such as document summarization [Lin and Bilmes, 2011], influence maximization [Kempe *et al.*, 2003] and sensor placement [Krause *et al.*, 2008], to name a few. This general NP-hard problem has been studied extensively, and many algorithms with bounded approximation guarantees have been proposed [Krause and Golovin, 2014]. For example, a well-known result is that when the objective function f is monotone submodular, the greedy algorithm, which iteratively adds one item with the largest marginal gain, can achieve the optimal approximation guarantee of $(1 - e^{-1})$ [Nemhauser *et al.*, 1978; Nemhauser and Wolsey, 1978].

However, in many practical applications such as recommendations in online shopping [McAuley *et al.*, 2015], paper reading [Shahaf *et al.*, 2012] and course learning [Parameswaran *et al.*, 2011], it is often desired to output a sequence instead of a subset. That is, the items cannot be treated independently, and the objective function f depends on the order of items. In this paper, we study the sequence selection problem, i.e., the goal is to select a sequence of at most k items that will maximize some given objective function f.

Note that the search space is exponentially larger by considering sequences instead of subsets, thus sequence selec-

tion can be harder than subset selection. To the best of our knowledge, only three pieces of studies have been reported on sequence selection. Alaei and Malekian [2010] first extended the notions of monotonicity and submodularity from subsets to sequences, and proved that for *sequence monotone submodular* objective functions, the GREEDY algorithm, which iteratively appends one item with the largest marginal gain to the end of the current sequence, can achieve a $(1 - e^{-1})$-approximation guarantee. When the objective function is relaxed to be *string monotone submodular* (which is weaker than sequence monotone submodular), Zhang et al. [2016] showed that GREEDY achieves a $(1/\sigma)(1 - e^{-\sigma})$-approximation guarantee, where σ is the curvature characterizing the degree of submodularity. Recently, Tschiatschek et al. [2017] considered a new class of objective functions, which does not necessarily satisfy the sequence submodular or string submodular property. They proved that GREEDY fails to achieve a constant approximation guarantee, and then proposed a new algorithm OMEGA with an approximation guarantee of $(1 - e^{-\frac{1}{2\Delta}})$, where $\Delta \geq 2$.

In this paper, we propose a Pareto Optimization method for Sequence Selection, briefly called POSEQSEL. It first reformulates the original problem as a bi-objective optimization problem that maximizes the given objective f and minimizes the sequence length simultaneously, then employs a randomized iterative procedure to solve it, and finally selects the best sequence satisfying the length constraint from the produced set of sequences. To theoretically investigate the performance of POSEQSEL, we consider each class of previously studied objective functions. We prove that POSEQSEL within polynomial time can always reach or improve the best known approximation guarantee. The concrete theoretical results are:

- For sequence monotone submodular objective functions, POSEQSEL achieves an approximation guarantee of $(1 - e^{-1})$ (**Theorem 1**), which reaches that of GREEDY [Alaei and Malekian, 2010].

- For string monotone submodular objective functions, POSEQSEL achieves an approximation guarantee of $(1/\sigma)(1 - e^{-\sigma})$ (**Theorem 2**), which reaches that of GREEDY [Zhang *et al.*, 2016].

- For a special class of objective functions (not necessarily

sequence or string submodular), POSEQSEL achieves an approximation guarantee of $1 - e^{-\frac{1}{2}}$ (**Theorem 3**), which is better than that of OMEGA [Tschiatschek *et al.*, 2017].

The experimental results also show the better performance of POSEQSEL over GREEDY and OMEGA.

The rest of the paper first introduces the studied problem, and then presents the proposed algorithm, its theoretical analysis and empirical study. Finally we conclude this paper.

2 Preliminaries

Let $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{Z}^+$ denote the set of reals, non-negative reals and non-negative integers, respectively. Given a finite set $V = \{v_1, v_2, \ldots, v_n\}$ of items, we study the functions $f : \mathcal{S} \to \mathbb{R}$ defined on sequences of items from V. A sequence is represented by $s \in \mathcal{S} = \{(s_1, s_2, \ldots, s_l) \mid s_i \in V, l \in \mathbb{Z}^+\}$, where $l = 0$ represents the empty sequence $\emptyset$. The sequence selection problem as presented in Definition 1 is to select a sequence s such that a given objective f is maximized with the constraint $|s| \le k$, where $|\cdot|$ denotes the length of a sequence. In this paper, we will use $s \oplus t$ to denote the concatenation of two sequences s and t, and represent a singleton sequence (v) by v for simplicity.

Definition 1 (Sequence Selection). *Given all items* $V = \{v_1, v_2, \ldots, v_n\}$, *an objective function* $f : \mathcal{S} \to \mathbb{R}$ *and a budget* $k \in \mathbb{Z}^+$, *it is to find a sequence of at most* k *items maximizing* f, *i.e.,*

$$\arg\max_{s \in \mathcal{S}} f(s) \quad s.t. \quad |s| \le k. \tag{1}$$

In [Alaei and Malekian, 2010], *sequence monotone submodular* objective functions were considered. As presented in Definition 2, the monotonicity and the submodularity (i.e., the diminishing return property) hold with the subsequence relationship (denoted by $\sqsubseteq_{seq}$). The GREEDY algorithm iteratively appends one item with the largest improvement on f to the end of the current sequence, and was proved able to achieve a $(1 - e^{-1})$-approximation guarantee [Alaei and Malekian, 2010]. In this paper, we always assume that monotone functions are normalized, i.e., $f(\emptyset) = 0$.

Definition 2 (Sequence Monotone Submodular [Alaei and Malekian, 2010]). *For two sequences* $s, t \in \mathcal{S}$, $s \sqsubseteq_{seq} t$ *if* s *is a subsequence of* t. *Then a sequence function* $f : \mathcal{S} \to \mathbb{R}$ *is sequence monotone submodular if*

(1) f *is sequence monotone, i.e.,* $\forall s \sqsubseteq_{seq} t : f(s) \le f(t)$;

(2) f *is sequence submodular, i.e.,*

$$\forall s \sqsubseteq_{seq} t, v \in V : f(s \oplus v) - f(s) \ge f(t \oplus v) - f(t).$$

In [Zhang *et al.*, 2016], a more general class of objective functions called *string monotone submodular* was considered. As presented in Definition 3, the monotonicity and the submodularity hold with the prefix relationship (denoted by $\sqsubseteq_{str}$). Note that prefix is a special case of substring, which is a special case of subsequence. Thus, string monotone submodular is weaker than sequence monotone submodular, i.e., a sequence monotone submodular function must be string monotone submodular. By utilizing the notion of *string curvature* as presented in Definition 4, it was

proved that GREEDY achieves an approximation guarantee of $(1/\sigma_{o,k}(f))(1 - e^{-\sigma_{o,k}(f)})$ [Zhang *et al.*, 2016], where o denotes an optimal sequence of Eq. (1).

Definition 3 (String Monotone Submodular [Zhang *et al.*, 2016]). *For two sequences* $s, t \in \mathcal{S}$, $s \sqsubseteq_{str} t$ *if* s *is a prefix of* t. *Then a sequence function* $f : \mathcal{S} \to \mathbb{R}$ *is string monotone submodular if*

(1) f *is string monotone, i.e.,* $\forall s \sqsubseteq_{str} t : f(s) \le f(t)$;

(2) f *is string submodular, i.e.,*

$$\forall s \sqsubseteq_{str} t, v \in V : f(s \oplus v) - f(s) \ge f(t \oplus v) - f(t).$$

Definition 4 (String Curvature [Zhang *et al.*, 2016]). *Let* $f : \mathcal{S} \to \mathbb{R}^+$ *be a string monotone submodular function. The string curvature of* f *with respect to a sequence* s *and a parameter* $m \ge 1$ *is*

$$\sigma_{s,m}(f) = \max_{t \in \mathcal{S}, 0 < |t| \le m} \left\{ 1 - \frac{f(t \oplus s) - f(s)}{f(t)} \right\}. \tag{2}$$

The string curvature characterizes the degree of string submodularity. When f is clear, we will use $\sigma_{s,m}$ for short. We then make the following observations:

Remark 1. $\sigma_{s,m}$ *increases with* m *for any* s; *and* $\sigma_{s,m} \ge 0$ *for any* s *and* $m \ge |s| - 1$, *since by letting* t *in Eq. (2) be* $(s_1, \ldots, s_{|s|-1})$, *we have*

$$f(t \oplus s) - f(t)$$
$$= \sum_{i=1}^{|s|} f(t \oplus (s_1, \ldots, s_i)) - f(t \oplus (s_1, \ldots, s_{i-1}))$$
$$\le \sum_{i=1}^{|s|} f((s_1, \ldots, s_i)) - f((s_1, \ldots, s_{i-1})) = f(s),$$

where the inequality is by the string submodularity of f *since for any* $1 \le i \le |s|$, $(s_1, \ldots, s_{i-1}) \sqsubseteq_{str} t \oplus (s_1, \ldots, s_{i-1})$.

In [Tschiatschek *et al.*, 2017], a special class of objective functions called *DAG monotone submodular* was considered. As presented in Definition 5, it is assumed that there exists a directed acyclic graph $G = (V, E)$ (if not counting self-cycles) capturing the ordered preferences among items, where an edge $(v_i, v_j) \in E$ means that there is an additional utility when selecting v_i before v_j; and the function f value of a sequence s can be determined by the set of edges induced by s and a corresponding monotone submodular set function $h : 2^E \to \mathbb{R}^+$. Note that a DAG monotone submodular function does not necessarily satisfy the sequence submodular or string submodular property, and the items in a sequence cannot be repeated here, i.e., $\mathcal{S} = \{(s_1, s_2, \ldots, s_l) \mid s_i \in V, l \in \mathbb{Z}^+, \forall i \ne j : s_i \ne s_j\}$ [Tschiatschek *et al.*, 2017]. For this special class of sequence functions, GREEDY fails to achieve a constant approximation guarantee, while the OMEGA algorithm, which greedily picks an edge instead of an item, can achieve an approximation guarantee of $1 - e^{-\frac{1}{2\Delta}}$ [Tschiatschek *et al.*, 2017], where $\Delta = \min\{\Delta_{\text{in}}, \Delta_{\text{out}}\}$, and Δ_{in}, Δ_{out} are the largest indegree and outdegree of the items in the graph G, respectively.

Definition 5 (DAG Monotone Submodular [Tschiatschek *et al.*, 2017]). *Given a directed acyclic graph (DAG)* $G = $

(V, E) *modeling the ordered preferences among items, a sequence function* $f : \mathcal{S} \to \mathbb{R}$ *is DAG monotone submodular if* $\forall s \in \mathcal{S}$,

$$f(s) = h(E(s)),$$

where $h : 2^E \to \mathbb{R}^+$ *is a monotone submodular set function, and* $E(s) = \{(s_i, s_j) \mid (s_i, s_j) \in E, i \leq j\}$ *is the set of edges induced by the sequence s.*

Note that the size of the search space is $\binom{n}{k}$ for selecting a subset with k items, while by considering sequences instead of subsets, the search space becomes much larger. The size is n^k if allowing repeated items in a sequence, and is $k!\binom{n}{k}$ if not, either of which is exponentially larger w.r.t. k.

3 The POSEQSEL Algorithm

In this section, we propose a new approach based on Pareto Optimization [Qian *et al.*, 2015a] for the Sequence Selection problem, briefly called POSEQSEL. Note that Pareto optimization is a recently emerged framework that uses bi-objective optimization as an intermediate step to solve single-objective optimization problems. It has been successfully applied to the subset selection problem [Friedrich and Neumann, 2015; Qian *et al.*, 2015a; 2017b], the multiset selection problem [Qian *et al.*, 2018] as well as the problem of selecting several pairwise disjoint subsets [Qian *et al.*, 2017a].

POSEQSEL reformulates the original problem Eq. (1) as a bi-objective maximization problem

$$\arg\max_{s \in \mathcal{S}} \quad \big(f_1(s), \ f_2(s)\big),$$

where $f_1(s) = \begin{cases} -\infty, & |s| \geq 2k \\ f(s), & \text{otherwise} \end{cases}$, $\quad f_2(s) = -|s|$.

That is, POSEQSEL maximizes the objective function f and minimizes the sequence length $|s|$ simultaneously. Note that the goal of setting f_1 to $-\infty$ is to exclude overly infeasible sequences, the length of which is at least $2k$.

In the bi-objective setting, both the two objective values have to be considered for comparing two sequences s and s'. s *weakly dominates* s' (i.e., s is *better* than s', denoted as $s \succeq s'$) if $f_1(s) \geq f_1(s')$ and $f_2(s) \geq f_2(s')$; s *dominates* s' (i.e., s is *strictly better* than s', denoted as $s \succ s'$) if $s \succeq s'$ and either $f_1(s) > f_1(s')$ or $f_2(s) > f_2(s')$. But if neither s is better than s' nor s' is better than s, they are *incomparable*.

The procedure of POSEQSEL is described in Algorithm 1. Starting from the empty sequence $\emptyset$ (line 1), it iteratively tries to improve the quality of the sequences in the archive P (lines 2-13). In each iteration, a new sequence s' is generated from an archived sequence s, which is randomly selected from the current P (lines 3-8); if s' is not dominated by any archived sequence (line 9), it will be added into P, and meanwhile those archived sequences weakly dominated by s' will be removed (line 10). Note that the domination-based comparison makes P always contain incomparable sequences.

To generate a new sequence s' from s (lines 4-8), it applies the insertion or deletion operator uniformly at random and repeats this process r times independently, where the number r is determined by the Poisson distribution with $\lambda = 1$. Note that for the empty sequence, the deletion operator will keep

---

Algorithm 1 POSEQSEL Algorithm

Input: all items $V = \{v_1, v_2, \ldots, v_n\}$, a sequence function $f : \mathcal{S} \to \mathbb{R}$ and a budget $k \in \mathbb{Z}^+$
Parameter: the number T of iterations
Output: a sequence $s \in \mathcal{S}$ with $|s| \leq k$
Process:

1: Let $P = \{\emptyset\}$ and $t = 0$.
2: **while** $t < T$ **do**
3:     Select a sequence s from P uniformly at random.
4:     r = a random number sampled from the Poisson distribution with $\lambda = 1$.
5:     $s' = s$.
6:     **for** $i = 1$ to r
7:         $s' = insert(s')$ or $delete(s')$, each with prob. $1/2$.
8:     **end for**
9:     **if** $\nexists t \in P$ such that $t \succ s'$ **then**
10:         $P = (P \setminus \{t \in P \mid s' \succeq t\}) \cup \{s'\}$.
11:     **end for**
12:     $t = t + 1$.
13: **end while**
14: **return** $\arg\max_{s \in P : |s| \leq k} f(s)$

---

it unchanged; when not allowing repeated items, the insertion operator will keep a sequence of length n unchanged. Such a reproduction behavior is inspired from the mutation operator in genetic programming [Durrett *et al.*, 2011; Qian *et al.*, 2015b].

Definition 6 (Insertion). *Given a sequence* $s \in \mathcal{S}$, *if allowing repeated items, the insertion operator first randomly selects an item* v *from* V *and then inserts* v *into a randomly chosen position of* s. *That is, a new sequence* $(s_1, \ldots, s_{i-1}, v, s_i, \ldots, s_{|s|})$ *is generated, where* v *is uniformly chosen from* V *at random, and* i *is uniformly chosen from* $\{1, 2, \ldots, |s| + 1\}$ *at random. If not allowing repeated items, the item* v *is randomly selected from* $V \setminus V(s)$, *where* $V(s)$ *denotes the set of items appearing in the sequence* s.

Definition 7 (Deletion). *Given a sequence* $s \in \mathcal{S}$, *the deletion operator deletes a randomly chosen item of* s. *That is, a new sequence* $(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_{|s|})$ *is generated, where* i *is uniformly chosen from* $\{1, 2, \ldots, |s|\}$ *at random.*

The number T of iterations of POSEQSEL could affect the quality of the produced sequence. Their relationship will be analyzed in the theoretical analysis, and we will use the theoretically derived T value in the experiments. After running T iterations, the best sequence (i.e., having the largest f value) satisfying the length constraint in P is returned (line 14).

4 Theoretical Analysis

In this section, we examine the theoretical performance of POSEQSEL for the sequence selection problem with each class of previously studied objective functions.

4.1 Sequence Monotone Submodular

We prove in Theorem 1 that for sequence monotone submodular objective functions, POSEQSEL can achieve a $(1 - e^{-1})$-approximation guarantee, which reaches the best known one

previously obtained by GREEDY [Alaei and Malekian, 2010]. Let OPT denote the optimal function value of Eq. (1), and let $\mathbb{E}[T]$ denote the average number T of iterations to reach the guarantee. The proof relies on Lemma 1, i.e., for any sequence s, there always exists one item, whose addition to the end of s brings an increment on f proportional to the current distance to the optimum.

Lemma 1. *Let $f : \mathcal{S} \to \mathbb{R}^+$ be a sequence monotone submodular function. For any sequence $s \in \mathcal{S}$, there exists one item $v \in V$ such that*

$$f(s \oplus v) - f(s) \geq (OPT - f(s))/k.$$

Proof. Let o be an optimal sequence of Eq. (1), i.e., $f(o) = OPT$. Then, for any $s \in \mathcal{S}$,

$$f(s \oplus o) - f(s) \qquad (3)$$
$$= \sum\nolimits_{i=1}^{|o|} f(s \oplus (o_1, \ldots, o_i)) - f(s \oplus (o_1, \ldots, o_{i-1}))$$
$$\leq \sum\nolimits_{i=1}^{|o|} f(s \oplus o_i) - f(s),$$

where the inequality is by the sequence submodularity of f since $s \sqsubseteq_{seq} s \oplus (o_1, \ldots, o_{i-1})$. Let $V(o)$ denote the set of items appearing in the sequence o, and let $v^* \in \arg\max_{v \in V(o)} f(s \oplus v)$. Then, we have

$$f(s \oplus v^*) - f(s) \geq (f(s \oplus o) - f(s))/|o|$$
$$\geq (f(s \oplus o) - f(s))/k, \qquad (4)$$

where the last inequality is by $|o| \leq k$ and $f(s \oplus o) - f(s) \geq 0$, since f is sequence monotone and $s \sqsubseteq_{seq} s \oplus o$. Since f is sequence monotone and $o \sqsubseteq_{seq} s \oplus o$, we also have

$$f(s \oplus o) \geq f(o) = OPT, \qquad (5)$$

and thus the lemma holds. $\square$

Theorem 1. *For the sequence selection problem with sequence monotone submodular objective functions,* POSE-QSEL *with $\mathbb{E}[T] \leq 2ek^2(k+1)n$ finds a sequence $s \in \mathcal{S}$ with $|s| \leq k$ and $f(s) \geq (1 - e^{-1}) \cdot OPT$.*

Proof. Let $J_{\max}$ denote the maximum value of $j \in \{0, 1, \ldots, k\}$ such that in the archive P of POSEQSEL (i.e., Algorithm 1), there exists a sequence s with $|s| \leq j$ and $f(s) \geq (1 - (1 - \frac{1}{k})^j) \cdot OPT$. That is,

$$J_{\max} = \max\{j \in \{0, 1, \ldots, k\} \mid \exists s \in P,$$
$$|s| \leq j \ \wedge \ f(s) \geq (1 - (1 - 1/k)^j) \cdot OPT\}.$$

We then only need to analyze the expected number of iterations until $J_{\max} = k$, since it implies that there exists one sequence s in P satisfying that $|s| \leq k$ and $f(s) \geq (1 - (1 - \frac{1}{k})^k) \cdot OPT \geq (1 - e^{-1}) \cdot OPT$.

The initial value of $J_{\max}$ is 0, since POSEQSEL starts from the empty sequence $\emptyset$. Assume that currently $J_{\max} = i < k$. Let s be a corresponding sequence with the value i, i.e.,

$$|s| \leq i \ \wedge \ f(s) \geq (1 - (1 - 1/k)^i) \cdot OPT. \qquad (6)$$

It is easy to see that $J_{\max}$ cannot decrease because deleting s from P (line 10 of Algorithm 1) implies that s is weakly dominated by the newly generated sequence s', which must satisfy that $|s'| \leq |s|$ and $f(s') \geq f(s)$. By Lemma 1, we know that appending a specific item to the end of s can generate a new sequence s' with $f(s') - f(s) \geq \frac{1}{k}(OPT - f(s))$. By applying Eq. (6) to this inequality, we get

$$f(s') \geq (1 - (1 - 1/k)^{i+1}) \cdot OPT.$$

Since $|s'| = |s| + 1 \leq i + 1$, s' will be included into P; otherwise, s' must be dominated by one sequence in P (line 9 of Algorithm 1), and this implies that $J_{\max}$ has already been larger than i, which contradicts with the assumption $J_{\max} = i$. After including s' into P, $J_{\max} \geq i + 1$. Let $P_{\max}$ denote the largest size of P during the run of POSE-QSEL. Thus, $J_{\max}$ can increase by at least 1 in one iteration with probability at least

$$\frac{1}{P_{\max}} \cdot \frac{1}{e} \cdot \frac{1}{2} \cdot \frac{1}{n} \cdot \frac{1}{|s| + 1} \geq \frac{1}{2e(i+1)nP_{\max}},$$

where $\frac{1}{P_{\max}}$ is a lower bound on the probability of selecting s in line 3 of Algorithm 1 due to uniform selection, $\frac{1}{e}$ is the probability of $r = 1$ (i.e, line 7 is implemented once) by the Poisson distribution with $\lambda = 1$, $\frac{1}{2}$ is the probability of performing insertion, and $\frac{1}{n} \cdot \frac{1}{|s|+1}$ is the probability of selecting a specific item from V and adding it to the end of s in insertion (i.e., Definition 6). Then, it needs at most $2e(i+1)nP_{\max}$ iterations in expectation to increase $J_{\max}$. We thus get that the expected number of iterations until $J_{\max} = k$ is at most

$$\sum\nolimits_{i=0}^{k-1} 2e(i+1)nP_{\max} = ek(k+1)nP_{\max}.$$

By the procedure of POSEQSEL, we know that the sequences maintained in P must be incomparable. Thus, each value of one objective can correspond to at most one sequence in P. Because the sequences with $|s| \geq 2k$ have $-\infty$ value on the first objective, they must be excluded from P. Thus, $P_{\max} \leq 2k$, and the expected number of iterations $\mathbb{E}[T]$ for finding a desired solution is at most $2ek^2(k+1)n$. $\square$

4.2 String Monotone Submodular

We prove in Theorem 2 that for string monotone submodular objective functions, POSEQSEL can achieve a $(1/\sigma_{o,k-1})(1 - e^{-\sigma_{o,k-1}})$-approximation guarantee, which reaches the best known one previously obtained by GREEDY [Zhang *et al.*, 2016]. Note that o denotes an optimal sequence of Eq. (1) (i.e., $f(o) = OPT$), and $\sigma_{o,k-1} \geq 0$ by Remark 1 since $|o| \leq k$. The proof of Theorem 2 is similar to that of Theorem 1. The main difference is that a different inductive inequality on f is used in the definition of the quantity $J_{\max}$. For concise illustration, we will mainly show the difference in the proof of Theorem 2.

Lemma 2. *Let $f : \mathcal{S} \to \mathbb{R}^+$ be a string monotone submodular function. For any sequence $s \in \mathcal{S}$, there exists one item $v \in V$ such that*

$$f(s \oplus v) - f(s) \geq (OPT - \sigma_{o,|s|} \cdot f(s))/k.$$

Proof. The proof is similar to that of Lemma 1. It is easy to verify that Eq. (3) still holds since the inequality in Eq. (3)

holds by the string submodularity of f and $s \sqsubseteq_{str} s \oplus (o_1, \ldots, o_{i-1})$; Eq. (4) still holds since $f(s \oplus o) - f(s) \geq 0$ holds by the string monotonicity of f and $s \sqsubseteq_{str} s \oplus o$. But Eq. (5) does not hold now, since o may not be a prefix of $s \oplus o$, and we cannot use the string monotonicity of f to derive $f(s \oplus o) \geq f(o)$. By the definition of string curvature (i.e., Definition 4), we get

$$\sigma_{o,|s|} = \max_{\substack{t \in \mathcal{S}, \\ 0 < |t| \leq |s|}} \left\{ 1 - \frac{f(t \oplus o) - f(o)}{f(t)} \right\} \geq 1 - \frac{f(s \oplus o) - f(o)}{f(s)},$$

which leads to $f(s \oplus o) \geq f(o) + (1 - \sigma_{o,|s|})f(s)$. By applying this inequality to Eq. (4), the lemma holds. Note that $\sigma_{o,|s|}$ is not defined for $s = \emptyset$, but the lemma still holds by applying $f(\emptyset) = 0$ to Eq. (4). $\quad\square$

Theorem 2. *For the sequence selection problem with string monotone submodular objective functions,* POSEQSEL *with* $\mathbb{E}[T] \leq 2ek^2(k+1)n$ *finds a sequence* $s \in \mathcal{S}$ *with* $|s| \leq k$ *and* $f(s) \geq (1/\sigma_{o,k-1})(1 - e^{-\sigma_{o,k-1}}) \cdot OPT.$

Proof. The proof is similar to that of Theorem 1. We use a different $J_{\max}$, which is defined as

$$J_{\max} = \max\{j \in \{0, 1, \ldots, k\} \mid \exists s \in P,$$
$$|s| \leq j \wedge f(s) \geq \frac{1}{\sigma_{o,k-1}} \left(1 - \left(1 - \frac{\sigma_{o,k-1}}{k} \right)^j \right) \cdot OPT \right\}.$$

It is easy to verify that $J_{\max} = k$ implies that the desired approximation guarantee is reached, since there must exist one sequence s in P satisfying that $|s| \leq k$ and $f(s) \geq \frac{1}{\sigma_{o,k-1}}(1 - (1 - \frac{\sigma_{o,k-1}}{k})^k) \cdot OPT \geq \frac{1}{\sigma_{o,k-1}}(1 - e^{-\sigma_{o,k-1}}) \cdot OPT$. Assume that currently $J_{\max} = i < k$ and s is a corresponding sequence, i.e., $|s| \leq i$ and

$$f(s) \geq \frac{1}{\sigma_{o,k-1}} \left(1 - \left(1 - \frac{\sigma_{o,k-1}}{k} \right)^i \right) \cdot OPT. \quad (7)$$

We then only need to show that appending one specific item to the end of s can generate a new sequence s' with $f(s') \geq \frac{1}{\sigma_{o,k-1}}(1 - (1 - \frac{\sigma_{o,k-1}}{k})^{i+1}) \cdot OPT$. By Lemma 2, we know that there exists one specific item, the addition of which to the end of s can generate a new sequence s', which satisfies that $f(s') - f(s) \geq \frac{1}{k}(OPT - \sigma_{o,|s|}f(s))$. By applying Eq. (7) to this inequality and using $\sigma_{o,|s|} \leq \sigma_{o,k-1}$ (since $|s| \leq i < k$ and $\sigma_{o,m}$ increases with m), we get

$$f(s') \geq \frac{1}{\sigma_{o,k-1}} \left(1 - \left(1 - \frac{\sigma_{o,k-1}}{k} \right)^{i+1} \right) \cdot OPT.$$

Thus, the theorem holds. $\quad\square$

4.3 DAG Monotone Submodular

For a DAG monotone submodular objective function f, we know from Definition 5 that there exists a DAG $G = (V, E)$ (not counting self-cycles) and a monotone submodular set function $h : 2^E \to \mathbb{R}^+$ such that for any $s \in \mathcal{S}$, $f(s) = h(E(s))$, where $E(s) = \{(s_i, s_j) \mid (s_i, s_j) \in E, i \leq j\}$ is the set of edges induced by s on G. In this case, the sequence submodular or string submodular property is not necessarily satisfied, and GREEDY fails to achieve a good approximation guarantee. Tschiatschek et al. [2017] then developed the

OMEGA algorithm by exploiting the DAG property of the graph G, i.e., for each set of items, its optimal ordering can be computed by first computing a topological ordering of G and then sorting the set of items according to that order. OMEGA obtains the best known approximation guarantee of $1 - e^{-\frac{1}{2\Delta}}$, where $\Delta = \min\{\Delta_{\text{in}}, \Delta_{\text{out}}\}$, and $\Delta_{\text{in}}, \Delta_{\text{out}}$ are the largest indegree and outdegree of the items in G, respectively.

Let $V(s)$ denote the set of items present in a sequence $s \in \mathcal{S}$, and let REORDER$(\cdot)$ denote the optimal sequence for a set of items. In the implementation of POSEQSEL, we also utilize the DAG property of the graph G: when computing $f(s)$, we directly use the f value of the optimal ordering for $V(s)$, i.e., $f(s) = h(E(\text{REORDER}(V(s))))$; when the algorithm terminates, we output REORDER$(V(s))$ instead of s. We prove in Theorem 3 that POSEQSEL can achieve an approximation guarantee of $(1 - e^{-\frac{1}{2}})$, which is better than the best known one, i.e., $1 - e^{-\frac{1}{2\Delta}}$ [Tschiatschek *et al.*, 2017], since Δ is usually much larger than 1. The proof relies on Lemma 3, that for any $s \in \mathcal{S}$, there always exist one or two items, the insertion of which into s can bring an improvement on f proportional to the current distance to the optimum.

Lemma 3. *Let* $f : \mathcal{S} \to \mathbb{R}^+$ *be a DAG monotone submodular function. For any sequence* $s \in \mathcal{S}$, *there exists one item* $v \in V \setminus V(s)$ *or two items* $u, v \in V \setminus V(s)$ *such that inserting* v *or* u, v *into any positions of* s *leads to a sequence* s' *with*

$$f(s') - f(s) \geq (OPT - f(s))/k.$$

Proof. The proof relies on an auxiliary set function $g : 2^E \to \mathbb{R}$. Let $V(X)$ denote the set of items covered by an edge set $X \subseteq E$. We define g as for any $X \subseteq E$, $g(X) = h(E(\text{REORDER}(V(X))))$. Note that g is monotone and submodular, as proved in Lemma 1 of [Tschiatschek *et al.*, 2017].

Let o be an optimal sequence of Eq. (1), and let $X^* \in \arg\min_{X \subseteq E, V(X) = V(o)} |X|$, i.e., X^* is the smallest edge set which covers the item set $V(o)$. It is easy to see that $|X^*| \leq k$, since $|o| \leq k$ and one edge can cover at least one item. Then, for any $s \in \mathcal{S}$,

$$g(X^*) - g(E(s)) \leq g(E(s) \cup X^*) - g(E(s))$$
$$\leq \sum_{e \in X^* \setminus E(s)} g(E(s) \cup e) - g(E(s)),$$

where the first inequality is by the monotonicity of g, i.e, $\forall X \subseteq Y, g(X) \leq g(Y)$, and the second is by the submodularity of g, i.e, $\forall X \subseteq Y, g(Y) - g(X) \leq \sum_{e \in Y \setminus X} g(X \cup e) - g(X)$. Let $e^* \in \arg\max_{e \in X^* \setminus E(s)} g(E(s) \cup e)$. Since $|X^* \setminus E(s)| \leq |X^*| \leq k$, we have

$$g(E(s) \cup e^*) - g(E(s)) \geq (g(X^*) - g(E(s)))/k.$$

By the definitions of f and g, we easily verify that for any $X \subseteq E$ and $s \in \mathcal{S}$, if $V(X) = V(s)$, $g(X) = f(s)$. Thus,

$$g(E(s)) = f(s) \wedge g(X^*) = f(o) = OPT,$$

since $V(E(s)) = V(s)$ and $V(X^*) = V(o)$. Let s' be any sequence with $V(s') = V(E(s) \cup e^*)$. Then,

$$f(s') = g(E(s) \cup e^*).$$

Since $V(s') = V(E(s) \cup e^*) = V(s) \cup V(e^*)$ and e^* introduces one or two new items, the lemma holds. $\quad\square$

Theorem 3. *For the sequence selection problem with DAG monotone submodular objective functions,* POSEQSEL *with* $\mathbb{E}[T] \leq 4ek^2n^2$ *finds a sequence* $s \in \mathcal{S}$ *with* $|s| \leq k$ *and* $f(s) \geq (1 - e^{-\frac{1}{2}}) \cdot OPT$.

Proof. The proof is similar to that of Theorem 1. We use a different $J_{\max}$, which is defined as

$$J_{\max} = \max\{j \in \{0, 1, \ldots, k\} \mid \exists s \in P,$$
$$|s| \leq j \ \wedge \ f(s) \geq (1 - (1 - 1/k)^{\lceil j/2 \rceil}) \cdot OPT\}.$$

We analyze the expected number of iterations until $J_{\max} \geq k - 1$, which implies that there exists one sequence s in P satisfying that $|s| \leq k$ and $f(s) \geq (1 - (1 - \frac{1}{k})^{\lceil (k-1)/2 \rceil}) \cdot OPT \geq (1 - e^{-\frac{k-1}{2k}}) \cdot OPT$, which is nearly $(1 - e^{-\frac{1}{2}}) \cdot OPT$ for sufficiently large k.

As the analysis in the proof of Theorem 1, $J_{\max}$ is initially 0 and never decreases. Assume that currently $J_{\max} = i < k - 1$ and s is a corresponding sequence, i.e., $|s| \leq i$ and

$$f(s) \geq (1 - (1 - 1/k)^{\lceil i/2 \rceil}) \cdot OPT. \tag{8}$$

By Lemma 3, we know that there exist one or two items, the insertion of which into s can generate a new sequence s', which satisfies that $f(s') - f(s) \geq \frac{1}{k}(OPT - f(s))$. By applying Eq. (8) to this inequality, we get

$$f(s') \geq (1 - (1 - 1/k)^{\lceil (i+2)/2 \rceil}) \cdot OPT.$$

Note that $|s'| \leq |s| + 2 \leq i + 2$. Thus, s' will be included into P, which makes $J_{\max} \geq i + 2$. The probability of inserting one specific item into s is obviously larger than that of inserting two specific items into s, which is at least

$$\frac{1}{P_{\max}} \cdot \frac{1}{2e} \cdot \frac{1}{2^2} \cdot \frac{2}{(n - |s|)(n - |s| - 1)} \geq \frac{1}{4en^2 P_{\max}},$$

where $\frac{1}{P_{\max}}$ is a lower bound on the probability of selecting s in line 3 of Algorithm 1, $\frac{1}{2e}$ is the probability of $r = 2$ (i.e, line 7 is implemented twice), $\frac{1}{2^2}$ is the probability of performing insertion twice, and $\frac{2}{(n-|s|)(n-|s|-1)}$ is the probability of selecting the two specific items for insertion. Note that $P_{\max} \leq 2k$. Thus, $J_{\max}$ can increase by at least 2 in one iteration with probability at least $\frac{1}{8ekn^2}$. We then get that the expected number of iterations until $J_{\max} \geq k - 1$ is at most

$$8ekn^2 \cdot \lceil (k-1)/2 \rceil \leq 4ek^2n^2.$$

Thus, the theorem holds. $\qquad\square$

5 Experiments

In this section, we investigate the empirical performance of POSEQSEL by synthetic experiments.

String Monotone Submodular. We consider the application of selecting a sequence of actions to maximize the expected fraction of accomplished tasks [Zhang *et al.*, 2016]. Given m tasks, n actions and a sequence $s = (s_1, s_2, \ldots, s_l)$ of actions, the objective function is $f(s) = \frac{1}{m} \sum_{i=1}^{m} (1 - \prod_{j=1}^{l} (1 - p_i^j(s_j)))$, where $p_i^j(s_j)$ is the probability of accomplishing task i by performing action s_j at stage j. We set
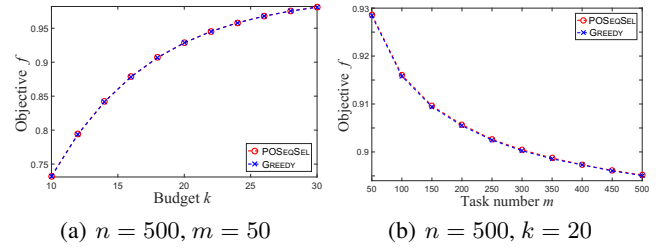

(a) $n = 500, m = 50$ (b) $n = 500, k = 20$

Figure 1: The comparison between POSEQSEL and GREEDY for string monotone submodular objective functions.
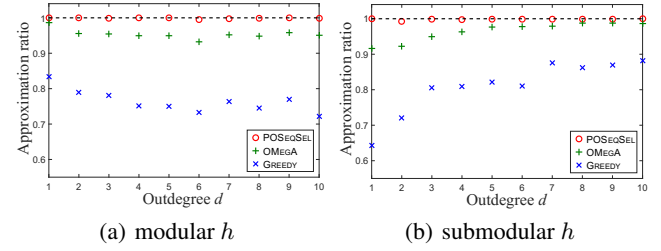

(a) modular h (b) submodular h

Figure 2: The comparison between POSEQSEL, OMEGA and GREEDY for DAG monotone submodular objective functions.

$m = 50$, $n = 500$, and each probability $p_i^j(s_j)$ is randomly sampled from $[0, 0.2]$. We compare POSEQSEL with the previous best algorithm GREEDY. The number T of iterations of POSEQSEL is set to $2ek^2(k + 1)n$ as suggested by Theorem 2. The budget k is set as $\{10, 12, \ldots, 30\}$. We randomly generate 50 problem instances, and report the average results, as shown in Figure 1(a). Similarly, Figure 1(b) shows the results for fixed $k = 20$ and varying $m \in \{50, 100, \ldots, 500\}$. We can observe that POSEQSEL and GREEDY achieve nearly the same performance, which verifies our theoretical analysis. The results are also consistent with that the optimal f value increases with k while decreases with m in expectation.

DAG Monotone Submodular. We use the same setting as in [Tschiatschek *et al.*, 2017]. The graph $G = (V, E)$ is constructed as follows: for each item $v_i \in V$, randomly select a subset of size $\min\{d, n - i\}$ from $\{v_{i+1}, \ldots, v_n\}$ and set an edge from v_i to each item in the selected subset and also to itself (i.e., a self-cycle). By assigning a weight to each edge, two set functions $h : 2^E \to \mathbb{R}$ are considered: $h(X) = \sum_{(v_i, v_j) \in X} w_{i,j}$ and $h(X) = \sum_{v_j \in V(X)} (1 - \prod_{(v_i, v_j) \in X} (1 - w_{i,j}))$, which are modular and submodular, respectively. Note that $V(X)$ denotes the item set covered by the edge set $X \subseteq E$. For modular h, each weight $w_{i,j}$ is randomly sampled from $[0, 1]$; for submodular h, each weight $w_{i,j}$ with $i < j$ is randomly sampled from $[0, 1]$ and each weight $w_{i,i}$ is randomly sampled from $[0, 0.1]$. We set $n = 30$ and $k = 5$. We compare POSEQSEL with the previous best algorithm OMEGA as well as GREEDY. The number T of iterations of POSEQSEL is set to $4ek^2n^2$ as suggested by Theorem 3. For each $d \in \{1, 2, \ldots, 10\}$, we randomly generate 50 problem instances, and report the average results. By computing the optimum using exhaustive enumeration, Figure 2 shows the approximation ratio of the output sequence of each algorithm. We can observe that POSEQSEL performs the best

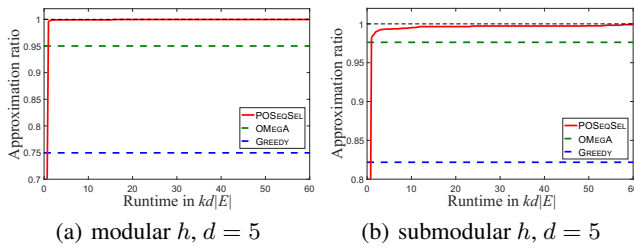(a) modular h, $d = 5$ (b) submodular h, $d = 5$

Figure 3: Performance v.s. runtime of POSEQSEL.

and almost finds the optimum, while GREEDY is the worst. Thus, these empirical results verify the theoretical analysis.

Runtime. Considering the runtime (in the number of objective function evaluations), GREEDY and OMEGA take the time in the order of kn and $kd|E|$ (where $|E|$ is the number of edges of G), respectively; POSEQSEL is set to use the theoretical upper bounds (i.e., the worst-case time), which are $2ek^2(k + 1)n$ and $4ek^2n^2$ for the two tested cases, respectively. We also empirically examine how effective POSE-QSEL is in practice. For the DAG monotone submodular case with $d = 5$, we plot the curve of the approximation ratio over the time for POSEQSEL and select GREEDY and OMEGA as the baselines. We can see from Figure 3 that POSEQSEL quickly obtains a better performance, which implies that POSEQSEL can be efficient in practice.

6 Conclusion

This paper studies the sequence selection problem, i.e., selecting a sequence with limited length that maximizes some objective function f. We propose a new method POSEQSEL, and prove that for any previously studied f, POSEQSEL can always reach or improve the best known approximation guarantee. Empirical results verify the theoretical analysis.

Acknowledgments

References

[Alaei and Malekian, 2010] S. Alaei and A. Malekian. Maximizing sequence-submodular functions and its application to online advertising. *arXiv:1009.4153*, 2010.

[Durrett *et al.*, 2011] G. Durrett, F. Neumann, and U.-M. O'Reilly. Computational complexity analysis of simple genetic programming on two problems modeling isolated program semantics. In *FOGA*, pages 69–80, 2011.

[Friedrich and Neumann, 2015] T. Friedrich and F. Neumann. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary Computation*, 23(4):543–558, 2015.

[Kempe *et al.*, 2003] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[Krause and Golovin, 2014] A. Krause and D. Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

[Krause *et al.*, 2008] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[Lin and Bilmes, 2011] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011.

[McAuley *et al.*, 2015] J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, pages 785–794, 2015.

[Nemhauser and Wolsey, 1978] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[Nemhauser *et al.*, 1978] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1):265–294, 1978.

[Parameswaran *et al.*, 2011] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems*, 29(4):20, 2011.

[Qian *et al.*, 2015a] C. Qian, Y. Yu, and Z.-H. Zhou. Subset selection by Pareto optimization. In *NIPS*, pages 1765–1773, 2015.

[Qian *et al.*, 2015b] C. Qian, Y. Yu, and Z.-H. Zhou. Variable solution structure can be helpful in evolutionary optimization. *Science China Information Sciences*, 58(11):1–17, 2015.

[Qian *et al.*, 2017a] C. Qian, J.-C. Shi, K. Tang, and Z.-H. Zhou. Constrained monotone k-submodular function maximization using multi-objective evolutionary algorithms with theoretical guarantee. *IEEE Transactions on Evolutionary Computation*, 2017.

[Qian *et al.*, 2017b] C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou. Subset selection under noise. In *NIPS*, pages 3562–3572, 2017.

[Qian *et al.*, 2018] C. Qian, Y. Zhang, K. Tang, and X. Yao. On multiset selection with size constraints. In *AAAI*, 2018.

[Shahaf *et al.*, 2012] D. Shahaf, C. Guestrin, and E. Horvitz. Metro maps of science. In *KDD*, pages 1122–1130, 2012.

[Tschiatschek *et al.*, 2017] S. Tschiatschek, A. Singla, and A. Krause. Selecting sequences of items via submodular maximization. In *AAAI*, pages 2667–2673, 2017.

[Zhang *et al.*, 2016] Z. Zhang, E. Chong, A. Pezeshki, and W. Moran. String submodular functions with curvature constraints. *IEEE Transactions on Automatic Control*, 61(3):601–616, 2016.