Maximizing Approximately k-Submodular Functions

Leqian Zheng[1] Hau Chan[2] Grigorios Loukides[3] Minming Li[1, 4]

[1]City University of Hong Kong, Hong Kong SAR, P. R. China
leqizheng2-c@my.cityu.edu.hk

[2]University of Nebraska-Lincoln, USA
hchan3@unl.edu

[3]King's College London, UK
grigorios.loukides@kcl.ac.uk

[4]City University of Hong Kong Shenzhen Research Institute, Shenzhen, P. R. China
minming.li@cityu.edu.hk [4]

SIAM International Conference on Data Mining (SDM) 2021

Preliminaries

- A **submodular function** $f : 2^V \to \mathbb{R}$ has a diminishing returns property [2]:

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y), \text{ for any } X, Y \in V$$

- A k-**submodular function** $f : (k+1)^V \to \mathbb{R}$ has a diminishing returns property w.r.t each subset when fixing the other subsets [4]:

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

for any $\boldsymbol{x}, \boldsymbol{y} \in (k+1)^V$ where

$$\boldsymbol{x} \sqcap \boldsymbol{y} = (X_1 \cap Y_1, X_2 \cap Y_2, \ldots, X_k \cap Y_k) \text{ and}$$
$$\boldsymbol{x} \sqcup \boldsymbol{y} = (X_1 \cup Y_1 \setminus (\bigcup_{i \in [k] \setminus \{1\}} X_i \cup Y_i), \ldots, X_k \cup Y_k \setminus (\bigcup_{i \in [k] \setminus \{k\}} X_i \cup Y_i))$$

- A **monotone** function $f : (k+1)^V \to \mathbb{R}$ satisfies:

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}), \text{ for any } \boldsymbol{x} \preceq \boldsymbol{y} \text{ (i.e., } X_i \subseteq Y_i, \forall i \in [k])$$

Motivation

- k-Submodular function optimization finds applications in
  - sensor placement, influence maximization, coupled feature selection and others
- In many cases, there is no access to the exact value of the function but only to noisy values of it
- We seek to answer two questions:
  - Q1. How to define an **approximately** k-submodular function?
  - Q2. What approximation guarantees can be obtained when maximizing such a function under **total size** (TS) or **individual size** (IS) constraints?

Overview of Contributions

- We address Q1 by introducing two natural definitions of an approximately k-submodular function as follows:

Definition (ε-approximately k-submodular or ε-approximately diminishing returns)

An function $F : (k+1)^V \to \mathbb{R}^+$ is ε-AS or ε-ADR if and only if for some small $\varepsilon > 0$, there exists a monotone *k-submodular* function f such that for any $\boldsymbol{x} \in (k+1)^V$, $u \notin \bigcup_{l \in [k]} X_l$, $u \in V$ and $i \in [k]$,

$$\varepsilon\text{-AS:}\ (1-\varepsilon) f(\boldsymbol{x}) \leq F(\boldsymbol{x}) \leq (1+\varepsilon) f(\boldsymbol{x}) \text{ or}$$

$$\varepsilon\text{-ADR:}\ (1-\varepsilon)\Delta_{u,i} f(\boldsymbol{x}) \leq \Delta_{u,i} F(\boldsymbol{x}) \leq (1+\varepsilon)\Delta_{u,i} f(\boldsymbol{x}),$$

where $\Delta_{u,i} f(\boldsymbol{x}) = f(X_1, \ldots, X_{i-1}, X_i \cup \{u\}, X_{i+1}, \ldots, X_k) - f(X_1, \ldots, X_k)$ and $\Delta_{u,i} F(\boldsymbol{x})$ is defined similarly.

- If F is ε-ADR, then F is ε-AS. However, the converse is not true.

Overview of Contributions

- By applying the greedy algorithms of [3] for TS and IS, we show that we can obtain the following approximation guarantees:

		ε-AS		ε-ADR
		F's Solution	f's Solution	F's Solution
$k=1$		$\frac{1}{1+\frac{4B\varepsilon}{(1-\varepsilon)^2}}\left(1-\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{2B}\left(1-\frac{1}{B}\right)^{B}\right)$ [1]	$\frac{1-\varepsilon}{1+\varepsilon}\left(1-\frac{1}{e}\right)$	$\left(1-e^{-\frac{(1-\varepsilon)}{(1+\varepsilon)}}\right)$
$k\geq 2$	TS	$\frac{(1-\varepsilon)^2}{2(1-\varepsilon+\varepsilon B)(1+\varepsilon)}$	$\frac{1-\varepsilon}{2(1+\varepsilon)}$	$\frac{1-\varepsilon}{2}$
	IS	$\frac{(1-\varepsilon)^2}{(3-3\varepsilon+2\varepsilon B)(1+\varepsilon)}$	$\frac{1-\varepsilon}{3(1+\varepsilon)}$	$\frac{1-\varepsilon}{3+\varepsilon}$

- Experiments on real data for the multi-type sensor allocation and multi-topic influence maximization problems show the impact of noise on the quality of the solutions and the effectiveness of our algorithms.

The k-Greedy-TS Algorithm

- The algorithm [3] essentially adds a single element with the highest marginal gain to one of the k subsets at each iteration without violating the TS constraint.

Algorithm 1: k-Greedy-TS (Total Size)

Input: a ε-approximately k-submodular function $F : (k+1)^V \mapsto \mathbb{R}^+$ and $B \in \mathbb{Z}^+$.

Output: a vector $\boldsymbol{x}$ with $|supp(\boldsymbol{x})| = B$.

$\boldsymbol{x} \leftarrow \mathbf{0}$;
for $j = 1$ **to** B **do**
  $(e, i) \leftarrow \arg\max_{e \in V \setminus supp(\boldsymbol{x}), i \in [k]} \Delta_{e,i} F(\boldsymbol{x})$;
  $\boldsymbol{x}(e) \leftarrow i$;
end

- It requires evaluating the function F $O(knB)$ times.

The k-`Greedy-TS` Algorithm

Lemma 1

For any $j \in [B]$, $\frac{1+\varepsilon}{1-\varepsilon} f(\boldsymbol{x}^{(j)}) - f(\boldsymbol{x}^{(j-1)}) \geq f(\boldsymbol{o}^{(j-1)}) - f(\boldsymbol{o}^{(j)})$.

Lemma 2

For any $j \in [B]$, it holds that $\frac{1}{1-\varepsilon}\left[F(\boldsymbol{x}^{(j)}) - F(\boldsymbol{x}^{(j-1)})\right] \geq \frac{1}{1+\varepsilon}\left[F(\boldsymbol{o}^{(j-1)}) - F(\boldsymbol{o}^{(j)})\right]$, if F is ε-ADR.

Theorem 1

For the total size constrained maximization problem, the k-`Greedy-TS` algorithm provides an approximation ratio of:

$$\frac{(1-\varepsilon)^2}{2(1-\varepsilon+\varepsilon B)(1+\varepsilon)}, \text{ if } F \text{ is } \varepsilon\text{-AS}$$

$$\frac{1-\varepsilon}{2}, \text{ if } F \text{ is } \varepsilon\text{-ADR}$$

for $k \geq 2$.

The k-Greedy-IS Algorithm

- The algorithm [3] essentially adds a single element with the highest marginal gain to one of the k subsets at each iteration without violating the IS constraints.

Algorithm 2: k-Greedy-IS (Individual Size)

Input: a ε-approximately k-submodular function $F : (k+1)^V \mapsto \mathbb{R}^+$ and $B_1, \cdots, B_k \in \mathbb{Z}^+$.

Output: a vector $\boldsymbol{x}$ with $|supp_i(\boldsymbol{x})| = B_i \ \forall i \in [k]$.

$\boldsymbol{x} \leftarrow \mathbf{0}$; $I \leftarrow [k]$;

while $I \neq \emptyset$ **do**

    $(e, i) \leftarrow \arg\max_{e \in V \setminus supp(\boldsymbol{x}), i \in I} \Delta_{e,i} F(\boldsymbol{x})$;

    $\boldsymbol{x}(e) \leftarrow i$;

    if $|supp_i(\boldsymbol{x}) = B_i|$ **then**

        $I \leftarrow I \setminus \{i\}$;

    end

end

- It requires evaluating the function F $O(kn \sum_{i \in [k]} B_i)$ times.

The k-`Greedy-IS` Algorithm

Lemma 3

For any $j \in [B]$, $2\left[\frac{1+\varepsilon}{1-\varepsilon} f(\boldsymbol{x}^{(j)}) - f(\boldsymbol{x}^{(j-1)})\right] \geq f(\boldsymbol{o}^{(j-1)}) - f(\boldsymbol{o}^{(j)})$.

Lemma 4

For any $j \in [B]$, $2\left[F(\boldsymbol{x}^{(j)}) - F(\boldsymbol{x}^{(j-1)})\right] \geq \frac{1-\varepsilon}{1+\varepsilon}\left[F(\boldsymbol{o}^{(j-1)}) - F(\boldsymbol{o}^{(j)})\right]$, if F is ε-ADR.

Theorem 2

For the individual size constrained maximization problem, the k-`Greedy-IS` algorithm provides an approximation ratio of:

$$\frac{(1-\varepsilon)^2}{(3-3\varepsilon+2\varepsilon B)(1+\varepsilon)}, \text{ if } F \text{ is } \varepsilon\text{-AS}$$

$$\frac{1-\varepsilon}{3+\varepsilon}, \text{ if } F \text{ is } \varepsilon\text{-ADR}$$

for $k \geq 2$.

Improved Approximation Ratios When f is Known

Definition

A function $F : (k+1)^V \to \mathbb{R}^+$ is ε-AS if and only if for some small $\varepsilon > 0$, there exists a monotone *k-submodular* function f such that for any $\boldsymbol{x} \in (k+1)^V$,

$$(1-\varepsilon)f(\boldsymbol{x}) \leq F(\boldsymbol{x}) \leq (1+\varepsilon)f(\boldsymbol{x})$$

Theorem 3

If there is an algorithm that provides an approximation ratio of α for maximizing f subject to constraint $\mathbb{X}$, then the same solution yields an approximation ratio of $\frac{1-\varepsilon}{1+\varepsilon}\alpha$ for maximizing F subject to constraint $\mathbb{X}$,

$$\frac{1-\varepsilon}{1+\varepsilon}(1-\frac{1}{e}),\ \textit{for } k=1$$

$$\frac{1-\varepsilon}{2(1+\varepsilon)},\ \textit{for TS constraint and } k \geq 2$$

$$\frac{1-\varepsilon}{3(1+\varepsilon)},\ \textit{for IS constraints and } k \geq 2$$

Experiments: Setup

For each $\boldsymbol{x} \in (k+1)^V$, the value of F of $\boldsymbol{x}$ should be generated such that $(1-\varepsilon)f(\boldsymbol{x}) \leq F(\boldsymbol{x}) \leq (1+\varepsilon)f(\boldsymbol{x})$.

- **Adversarial Generation (AG):** First run a greedy algorithm on f and obtain its solution $\boldsymbol{x_f}$, then let $F(\boldsymbol{x_f}) = (1+\varepsilon)f(\boldsymbol{x_f})$ which yields higher weight to f's solution. For the remaining $\boldsymbol{x}$, we let $F(\boldsymbol{x}) = \xi(\boldsymbol{x}) \cdot f(\boldsymbol{x})$ where $\xi(\boldsymbol{x}) \xleftarrow{\$} [1-\varepsilon, 1]$.
- **Max Generation (MaxG):** $F(\boldsymbol{x}) = \xi(\boldsymbol{x}) \cdot f(\boldsymbol{x})$, where $\xi(\boldsymbol{x}) = \max_{x \in supp(\boldsymbol{x})} \xi(x)$ and $\xi(x) \in [1-\varepsilon, 1]$. Thus, we weigh f with the maximum value of noise over the elements of $\boldsymbol{x}$.
- **Mean Generation (MeanG):** $F(\boldsymbol{x}) = \xi(\boldsymbol{x}) \cdot f(\boldsymbol{x})$, where $\xi(\boldsymbol{x}) = \frac{\sum_{x \in supp(\boldsymbol{x})} \xi(x)}{|supp(\boldsymbol{x})|}$ and $\xi(x) \in [1-\varepsilon, 1]$. Thus, we weigh f with the expected value of noise over $\boldsymbol{x}$.

Experiments: Multi-type Sensor Placement

- **The problem:** Install a vector $\boldsymbol{x} = (X_1, \cdots, X_k)$ of sensors of k types, one in each location, so that $\tilde{H}(\boldsymbol{x}) = \xi(\boldsymbol{x}) \cdot H(\boldsymbol{x})$ is maximum:
  - $H(\boldsymbol{x})$ is the entropy of $\boldsymbol{x}$; it quantifies the uncertainty of measurements.
  - $\tilde{H} = \xi(\boldsymbol{x}) \cdot H(\boldsymbol{x})$ is measured entropy of $\boldsymbol{x}$ with noise function $\xi(\boldsymbol{x})$ generated by our AG, MaxG, or MeanG method.
  - Gr-$\tilde{H}(\boldsymbol{x})$ or Gr-$H(\boldsymbol{x})$ is the measured entropy of $\boldsymbol{x}$ by applying k-`Greedy-TS` algorithm on $\tilde{H}$ or H, resp.
- **The dataset:** Intel Lab dataset[1]. About 2.3M values from 54 sensors of three types, collecting temperature, humidity, and light values.
- **The baseline:**
  - `Random`, which allocates sensors of any type randomly to locations.

[1] `http://db.csail.mit.edu/labdata/labdata.html`

Experiments: Multi-type Sensor Placement

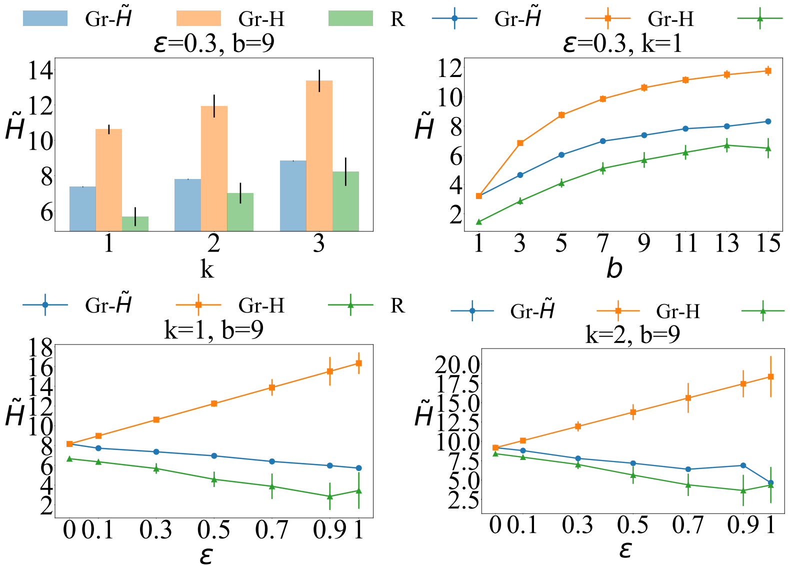

Figure: $\tilde{H}$ in AG setting

Experiments: Multi-type Sensor Placement

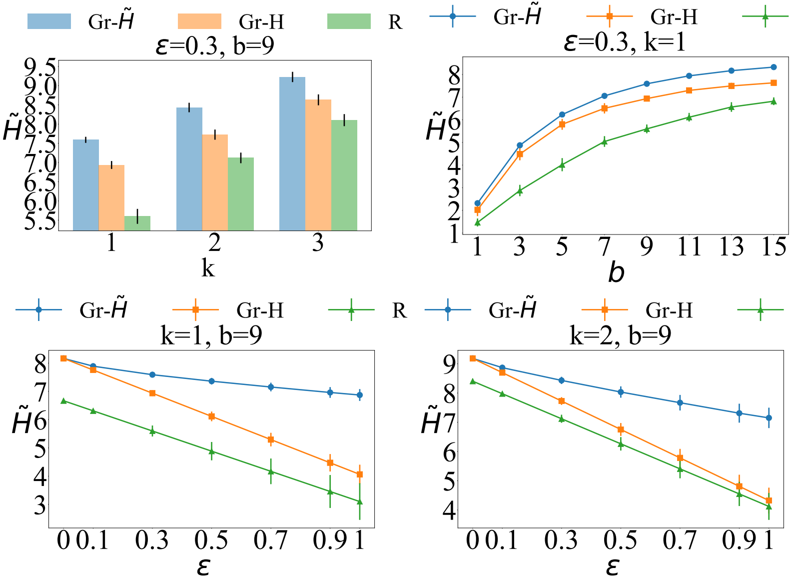

Figure: $\tilde{H}$ in MeanG setting

Experiments: Multi-topic Influence Maximization

- **The problem:** Find a vector $\boldsymbol{x} = (X_1, \cdots, X_k)$ of k user subsets, so that, when each subset starts a viral marketing campaign on a different topic, $\tilde{I}(\boldsymbol{x}) = \xi(\boldsymbol{x}) \cdot I(\boldsymbol{x})$ is maximum:
  - $I(\boldsymbol{x})$ is the expected number of influenced users about all topics under the k-topic IC model [3].
  - $\tilde{I}(\boldsymbol{x})$ is the measured spread with noise function $\xi(\boldsymbol{x})$ generated by our AG, MaxG, or MeanG method.
  - Gr-$\tilde{I}(\boldsymbol{x})$ or Gr-$I(\boldsymbol{x})$ is the measured spread of $\boldsymbol{x}$ by applying k-`Greedy-IS` algorithm on $\tilde{I}$ or I, resp.
- **The dataset:** Digg social network[2]. A node represents a user and each edge (u, v) represents that user u can watch the activity of v.
- **The baselines:**
  - `Random`, similar to that of sensor placement, and
  - `Degree`, which sorts all nodes in decreasing order of out-degree and assigns each of them to a random topic.

[2] `http://www.isi.edu/~lerman/downloads/digg2009.html`

Experiments: Multi-topic Influence Maximization

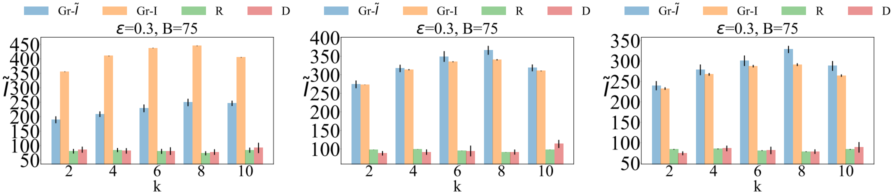

Figure: $\tilde{I}$ for varying k in: (a) AG, (b) MeanG, and (c) MaxG setting.

Conclusions

- We introduced the notions of approximately k-submodular and approximately diminishing returns functions.
- We showed that k-`Greedy-TS` and k-`Greedy-TS` algorithms have reasonable approximation ratios for ε-AS or ε-ADR function F subject to total size and individual size constraints.
- We demonstrated the effectiveness of the algorithms in sensor placement and influence maximization[3].

Thank you!

[3] `https://github.com/55199789/approx_kSubmodular.git`

References

[1] Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *NIPS*, pages 3045–3053. Curran Associates, Inc., 2016.

[2] Anna Huber and Vladimir Kolmogorov. Towards minimizing k-submodular functions. In *COCOA*, pages 451–462, 2012.

[3] Naoto Ohsaka and Yuichi Yoshida. Monotone k-submodular function maximization with size constraints. In *NIPS*, pages 694–702, 2015.

[4] Justin Ward and Stanislav Živný. Maximizing k-submodular functions and beyond. *ACM Trans. Algorithms*, 12(4):47:1–47:26, 2016.