Non-monotone Adaptive Submodular Maximization

Alkis Gotovos  
ETH Zurich

Amin Karbasi  
Yale University

Andreas Krause  
ETH Zurich
Many AI problems boil down to selecting a number of elements from a large set of options
Motivation

- Many AI problems boil down to selecting a number of elements from a large set of options

- Sequentially make smart choices based on past observations
Motivation

- Many AI problems boil down to selecting a number of elements from a large set of options

- Sequentially make smart choices based on past observations
Find classes of objective functions that are amenable to efficient sequential optimization with theoretical approximation guarantees
Non-monotone Adaptive Submodular Maximization
Objective

Ground set $V = \{a; b; c; d\}$

Objective function $f: 2^V \rightarrow \mathbb{R}_{\geq 0}$

- $f(\{d\} \cup \{g\}) = 4$
- $f(\{c; d\} \cup \{g\}) = 5$
Objective

- Ground set $V = \{a, b, c, d\}$
Objective

- Ground set $V = \{a, b, c, d\}$
- Objective function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$
Objective

- Ground set $V = \{a, b, c, d\}$

- Objective function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$

- $f(\{d\}) = 4$
Objective

- **Ground set** \( V = \{a, b, c, d\} \)

- **Objective function** \( f : 2^V \rightarrow \mathbb{R}_{\geq 0} \)

- \( f(\{d\}) = 4 \)

- \( f(\{c, d\}) = 5 \)
Objective

- $f$ is monotone
Objective

- $f$ is monotone
- $f$ is submodular
Objective

- $f$ is monotone
- $f$ is submodular
- Benefit of visiting $c$, given that...

Non-monotone Adaptive Submodular Maximization

Alkis Gotovos 15
Objective

- $f$ is monotone

- $f$ is submodular

- Benefit of visiting $c$, given that...
  - ...it is the first place we visit:
    \[ f(\{c\}) = 3 \]
Objective

- $f$ is monotone

- $f$ is submodular

- Benefit of visiting $c$, given that...
  - ...it is the first place we visit:
    $$f(\{c\}) = 3$$
  - ...we have already visited $d$:
    $$f(\{c, d\}) - f(\{d\}) = 5 - 4 = 1$$
Unconstrained problem:

\[
\text{maximize } f(S)
\]
Monotone submodular maximization

- Unconstrained problem:

\[
\text{maximize } f(S) \quad \rightarrow \quad \text{Trivial } \quad \text{OPT} = f(V)
\]
Monotone submodular maximization

- **Unconstrained problem:**

  \[
  \text{maximize } f(S) \quad \rightarrow \quad \text{Trivial } \quad \text{OPT} = f(V)
  \]

- **Cardinality-constrained problem:**

  \[
  \begin{align*}
  \text{maximize} & \quad f(S) \\
  \text{subject to} & \quad |S| \leq k
  \end{align*}
  \]
Unconstrained problem:

\[
\text{maximize } f(S) \quad \rightarrow \quad \text{Trivial } \quad \text{OPT } = f(V)
\]

Cardinality-constrained problem:

\[
\begin{align*}
\text{maximize } & \quad f(S) \\ 
\text{subject to } & \quad |S| \leq k
\end{align*} \quad \rightarrow \quad \text{NP-hard}
\]
Monotone submodular maximization

- **Unconstrained problem:**
  \[
  \text{maximize} \quad f(S) \quad \rightarrow \quad \text{Trivial} \quad \text{OPT} = f(V)
  \]

- **Cardinality-constrained problem:**
  \[
  \text{maximize} \quad f(S) \quad \text{subject to} \quad |S| \leq k \quad \rightarrow \quad \text{NP-hard}
  \]

- **More general constraints:** matroid, knapsack, etc.
Monotone submodular maximization

- Unconstrained problem:

  \[
  \text{maximize } f(S) \quad \rightarrow \quad \text{Trivial } \quad \text{OPT} = f(V)
  \]

- Cardinality-constrained problem:

  \[
  \begin{align*}
  \text{maximize } & \quad f(S) \\
  \text{subject to } & \quad |S| \leq k
  \end{align*}
  \quad \rightarrow \quad \text{NP-hard}
  \]

- More general constraints: matroid, knapsack, etc.
Greedy

\[ I_k = 2 \]

\[ I_{S_0} = \emptyset \]

\[ f(S_0) = 0 \]

\[ I_{S_1} = f_{d; a} \]

\[ f(S_1) = 4 \]

\[ I_{S_2} = f_{d; a} \]

\[ f(S_2) = 6 \]
Greedy

\[ k = 2 \]

\[ S_0 = \emptyset \]
\[ f(S_0) = 0 \]
\[ S_1 = f_d \]
\[ f(S_1) = 4 \]
\[ S_2 = f_d; a \]
\[ f(S_2) = 6 \]
Greedy

- $k = 2$

- $S_0 = \emptyset \rightarrow f(S_0) = 0$
$k = 2$

$S_0 = \emptyset \implies f(S_0) = 0$

$S_1 = \{d\} \implies f(S_1) = 4$
Greedy

- $k = 2$
- $S_0 = \emptyset \rightarrow f(S_0) = 0$
- $S_1 = \{d\} \rightarrow f(S_1) = 4$
- $S_2 = \{d, a\} \rightarrow f(S_2) = 6$

Non-monotone Adaptive Submodular Maximization

Alkis Gotovos 28
Theorem [Nemhauser et al., 1978]

If $f$ is monotone submodular, then greedy gives a $(1 - 1/e)$-approximation.
Birdwatching with costs
Non-monotonicity

\[ g(A) = f(A) - c(A) \]

- \( g(A) \) is a monotone submodular function
- \( f(A) \) is a monotone submodular function
- \( c(A) \) is a cost term

Greedy has no guarantees for non-monotone functions

Introduce randomization: random greedy algorithm
Non-monotonicity

\[ g(A) = f(A) - c(A) \]

- monotone submodular
- cost term

- Greedy has no guarantees for non-monotone functions
Non-monotonicity

\[ g(A) = f(A) - c(A) \]

- **g(A)**: monotone submodular cost term

- **Greedy has no guarantees for non-monotone functions**

- **Introduce randomization** \( \rightarrow \) **random greedy algorithm**
Random greedy

**Theorem** [Buchbinder *et al.*, 2014]

If $f$ is submodular, then random greedy gives a $\left(\frac{1}{e}\right)$-approximation*.

* In expectation over the randomness of the algorithm.
Random greedy

**Theorem** [Buchbinder *et al.*, 2014]

If $f$ is submodular, then random greedy gives a $(1/e)$-approximation*.

If $f$ is also monotone, then random greedy gives a $(1 - 1/e)$-approximation*.

* In expectation over the randomness of the algorithm.
Stochastic birdwatching

Non-monotone Adaptive Submodular Maximization
Stochastic birdwatching

Non-monotone Adaptive Submodular Maximization
Adaptivity

- Non-adaptive: choose set of locations in advance without looking at outcomes
Adaptivity

- Non-adaptive: choose set of locations in advance without looking at outcomes
- Adaptive: sequentially make choices based on past outcomes
Adaptivity

- Non-adaptive: choose set of locations in advance without looking at outcomes
- Adaptive: sequentially make choices based on past outcomes
- Monotonicity and submodularity $\leadsto$ adaptive monotonicity and adaptive submodularity
Adaptivity

- Non-adaptive: choose set of locations in advance without looking at outcomes
- Adaptive: sequentially make choices based on past outcomes
- Monotonicity and submodularity \(\rightarrow\) adaptive monotonicity and adaptive submodularity
- Greedily select the most promising location in conditional expectation \(\rightarrow\) adaptive greedy algorithm
Adaptive greedy

**Theorem** [Golovin and Krause, 2011]

If $f$ is adaptive monotone submodular, then adaptive greedy gives a $(1 - 1/e)$-approximation*.

* In expectation over the randomness of the environment.
<table>
<thead>
<tr>
<th></th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-monotone</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-adaptive</td>
<td>Adaptive</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>Monotone</td>
<td>Greedy</td>
<td>(1 − 1/(e))</td>
</tr>
<tr>
<td>Non-monotone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## What’s missing?

<table>
<thead>
<tr>
<th>Monotone</th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Greedy</td>
<td>Random greedy</td>
</tr>
<tr>
<td></td>
<td>$(1 - 1/e)$</td>
<td>$(1/e)$</td>
</tr>
<tr>
<td>Non-monotone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Non-monotone Adaptive Submodular Maximization
What’s missing?

<table>
<thead>
<tr>
<th></th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone</td>
<td>Greedy $(1 - 1/e)$</td>
<td>Adaptive greedy $(1 - 1/e)$</td>
</tr>
<tr>
<td>Non-monotone</td>
<td>Random greedy $(1/e)$</td>
<td></td>
</tr>
</tbody>
</table>
## What’s missing?

<table>
<thead>
<tr>
<th></th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monotone</strong></td>
<td>Greedy $(1 - 1/e)$</td>
<td>Adaptive greedy $(1 - 1/e)$</td>
</tr>
<tr>
<td><strong>Non-monotone</strong></td>
<td>Random greedy $(1/e)$</td>
<td>?</td>
</tr>
</tbody>
</table>
What's missing?
How do we maximize a non-monotone adaptive submodular function subject to a cardinality constraint?
How do we maximize a non-monotone adaptive submodular function subject to a cardinality constraint?

Adaptive random greedy
Adaptive random greedy

Theorem [Our contribution]

If $f$ is adaptive submodular, then adaptive random greedy gives a $(1/e)$-approximation*.

* In expectation over the randomness of the algorithm and the environment.
Theorem [Our contribution]

If $f$ is adaptive submodular, then adaptive random greedy gives a $(1/e)$-approximation*.

If $f$ is also adaptive monotone, then adaptive random greedy gives a $(1 - 1/e)$-approximation*.

* In expectation over the randomness of the algorithm and the environment.
<table>
<thead>
<tr>
<th></th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monotone</td>
<td>Greedy $(1 - 1/e)$</td>
<td>Adaptive greedy $(1 - 1/e)$</td>
</tr>
<tr>
<td>Non-monotone</td>
<td>Random greedy $(1/e)$</td>
<td>?</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Non-adaptive</th>
<th>Adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monotone</strong></td>
<td>Greedy $(1 - 1/e)$</td>
<td>Adaptive greedy $(1 - 1/e)$</td>
</tr>
<tr>
<td><strong>Non-monotone</strong></td>
<td>Random greedy $(1/e)$</td>
<td>Adaptive random greedy $(1/e)$</td>
</tr>
</tbody>
</table>
More in our poster! (Panel 40)

- Details on algorithm
- Classes of non-monotone objectives
- Experimental evaluation on social networks