Fast Maximization of Non-Submodular, Monotonic Functions on the Integer Lattice

Alan Kuhnle, J. David Smith, Victoria G. Crawford, My T. Thai

University of Florida, kuhnle@ufl.edu

ICML
July 12, 2018
Integer Lattice

- A natural extension of finite sets $S$

Example
A natural extension of finite sets $S$

- Sets can be represented as elements from $\{0, 1\}^S$
A natural extension of finite sets $S$

- Sets can be represented as elements from $\{0, 1\}^S$
- The integer lattice: $\mathbb{N}^S$

Example
A natural extension of finite sets $S$
- Sets can be represented as elements from $\{0, 1\}^S$
- The integer lattice: $\mathbb{N}^S$

Example
- Sensor placement
Integer Lattice

- A natural extension of finite sets $S$
  - Sets can be represented as elements from $\{0, 1\}^S$
  - The integer lattice: $\mathbb{N}^S$

Example

- Sensor placement
  - placement ($\{0, 1\}$) to power level in $\{0, \ldots, b\}$
Integer Lattice

- A natural extension of finite sets $S$
  - Sets can be represented as elements from $\{0, 1\}^S$
  - The integer lattice: $\mathbb{N}^S$

Example

- Sensor placement
  - placement ($\{0, 1\}$) to power level in $\{0, \ldots, b\}$
- Influence Maximization
A natural extension of finite sets $S$
  
  - Sets can be represented as elements from $\{0, 1\}^S$
  - The integer lattice: $\mathbb{N}^S$

Example

- Sensor placement
  
  - placement ($\{0, 1\}$) to power level in $\{0, \ldots, b\}$

- Influence Maximization
  
  - free products $\rightarrow x\%$ off coupon
Problem definition

Maximization subject to a cardinality constraint (MCC):

Definition

Let $f : \mathbb{N}^S \to \mathbb{R}^+$ be a non-negative and monotonic\(^1\) function with $f(\mathbf{0}) = 0$.

\[^1\text{for all } \mathbf{v} \leq \mathbf{w} \text{ (coordinate-wise)}, f(\mathbf{v}) \leq f(\mathbf{w})\]
Problem definition

Maximization subject to a cardinality constraint (MCC):

**Definition**

- Let $f : \mathbb{N}^S \to \mathbb{R}^+$ be a non-negative and monotonic\(^1\) function with $f(0) = 0$.
- Then determine
  $$\max_{\|w\|_1 \leq k} f(w),$$
  \[(MCC)\]
  where $w = (w_s)_{s \in S} \in \mathbb{N}^S$, $\|w\|_1 = \sum_{s \in S} |w_s|$, $k \in \mathbb{N}$.

\(^1\)for all $v \leq w$ (coordinate-wise), $f(v) \leq f(w)$.
Measures of submodularity

- diminishing-return (DR) ratio $\gamma_d$
Measures of submodularity

- **diminishing-return (DR) ratio** $\gamma_d$
  - maximum value such that

  
  $$
  \gamma_d \left( f(w + s) - f(w) \right) \leq f(v + s) - f(v),
  $$

  where $v \leq w$ and $s$ is step in $s$th direction

  
  $0 \leq \gamma_d \leq \gamma_s \leq 1$. Both take value 1 iff $f$ is DR submodular.
Measures of submodularity

- **diminishing-return (DR) ratio** $\gamma_d$
  - maximum value such that
    \[
    \gamma_d (f(w + s) - f(w)) \leq f(v + s) - f(v),
    \]
    where $v \leq w$ and $s$ is step in $s$th direction

- **weak DR ratio** $\gamma_s$

\[
0 \leq \gamma_d \leq \gamma_s \leq 1. \text{ Both take value 1 iff } f \text{ is DR submodular.}
\]
Measures of submodularity

- **diminishing-return (DR) ratio** $\gamma_d$
  - maximum value such that
  \[
  \gamma_d (f(w + s) - f(w)) \leq f(v + s) - f(v),
  \]
  where $v \leq w$ and $s$ is step in $s$th direction

- **weak DR ratio** $\gamma_s$
  - maximum value such that for all $v, w$, such that $v \leq w$,
  \[
  \gamma_s (f) (f(w) - f(v)) \leq \sum_{s \in \{w - v\}} f(v + s) - f(v).
  \]
Measures of submodularity

- **diminishing-return (DR) ratio** $\gamma_d$
  - maximum value such that
    \[
    \gamma_d (f(w + s) - f(w)) \leq f(v + s) - f(v),
    \]
    where $v \leq w$ and $s$ is step in $s$th direction

- **weak DR ratio** $\gamma_s$
  - maximum value such that for all $v, w$, such that $v \leq w$,
    \[
    \gamma_s (f)(f(w) - f(v)) \leq \sum_{s \in \{w-v\}} f(v + s) - f(v).
    \]

- $0 \leq \gamma_d \leq \gamma_s \leq 1$. Both take value 1 iff $f$ is DR submodular.
Our contributions

- ThresholdGreedy

Approximation ratio \(1 - e^{-\gamma d - \eta s}\) for any \(\eta > 0\)

Query complexity \(O(n \log k)\)

FastGreedy

Approximation ratio of \(1 - e^{-\beta \gamma s - \eta}\)

\(\beta\) is at least \(\gamma d\) and is computed by the algorithm

Uses non-submodularity to decrease its runtime in practice
Our contributions

- **ThresholdGreedy**
  - Approximation ratio $(1 - e^{-\gamma d \gamma s} - \eta)$ for any $\eta > 0$

- **FastGreedy**
  - Approximation ratio of $(1 - e^{-\gamma d \gamma s} - \eta)$

- $\beta$ is at least $\gamma d$ and is computed by the algorithm

- Uses non-submodularity to decrease its runtime in practice
Our contributions

▶ **ThresholdGreedy**
  ▶ Approximation ratio \((1 - e^{-\gamma d \gamma s} - \eta)\) for any \(\eta > 0\)
  ▶ Query complexity \(O(n \log k)\)
Our contributions

- **ThresholdGreedy**
  - Approximation ratio \(1 - e^{-\gamma d \gamma s} - \eta\) for any \(\eta > 0\)
  - Query complexity \(O(n \log k)\)

- **FastGreedy**

  \(\beta\) is at least \(\gamma d\) and is computed by the algorithm

Uses non-submodularity to decrease its runtime in practice
Our contributions

- **ThresholdGreedy**
  - Approximation ratio \((1 - e^{-\gamma d \gamma_s} - \eta)\) for any \(\eta > 0\)
  - Query complexity \(O(n \log k)\)

- **FastGreedy**
  - Approximation ratio of \((1 - e^{-\beta \gamma_s} - \eta)\)
    - \(\beta \) is at least \(\gamma d\) and is computed by the algorithm
    - Uses non-submodularity to decrease its runtime in practice
Our contributions

- **ThresholdGreedy**
  - Approximation ratio \((1 - e^{-\gamma_d \gamma_s} - \eta)\) for any \(\eta > 0\)
  - Query complexity \(O(n \log k)\)

- **FastGreedy**
  - Approximation ratio of \((1 - e^{-\beta \gamma_s} - \eta)\)
  - \(\beta\) is at least \(\gamma_d\) and is computed by the algorithm
Our contributions

- **ThresholdGreedy**
  - Approximation ratio \( (1 - e^{-\gamma_d \gamma_s} - \eta) \) for any \( \eta > 0 \)
  - Query complexity \( O(n \log k) \)

- **FastGreedy**
  - Approximation ratio of \( (1 - e^{-\beta \gamma_s} - \eta) \)
  - \( \beta \) is at least \( \gamma_d \) and is computed by the algorithm
  - Uses non-submodularity to decrease its runtime in practice
ThresholdGreedy

- Operates by considering decreasing thresholds for the marginal gain
ThresholdGreedy

- Operates by considering decreasing thresholds for the marginal gain
  - For each threshold $\tau$, adds copies of $s \in S$ such that the average marginal gain exceeds $\tau$
ThresholdGreedy

\[ S = \{0, 1, \ldots, 4\} , \ x = (0, 0, 0, 0, 0) \]

\[ \delta_0(x) \geq \tau ? \]
$S = \{0, 1, \ldots, 4\}, \ x = (0, 0, 0, 0, 0)$

$\delta_1(x) \geq \tau$?
Marginal gain, $\delta_i(x)$

ThresholdGreedy

$\delta_2(x) \geq \tau$?
$S = \{0, 1, \ldots, 4\}, \ x = (0, 0, 0, 0, 0)$

Yes, add it to $x$
ThresholdGreedy

$S = \{0, 1, \ldots, 4\}$, $x = (0, 0, 1, 0, 0)$

Marginal gain, $\delta_i(x)$

ThresholdGreedy

$\tau = 8$
ThresholdGreedy

If DR submodular, marginal gains decrease.

\[ S = \{0, 1, \ldots, 4\}, \ x = (0, 0, 1, 0, 0) \]
Marginal gain, $\delta_i(x)$

$S = \{0, 1, \ldots, 4\}$, $x = (0, 0, 1, 0, 0)$
\( S = \{0,1,\ldots,4\} \), \( x = (0,0,1,0,0) \)
\[ S = \{0, 1, \ldots, 4\}, \quad x = (0, 0, 1, 0, 0) \]
If not submodular, marginal gains may increase...
Marginal gain, $\delta_i(x)$

ThresholdGreedy

$S = \{0, 1, \ldots, 4\}$, $x = (0, 0, 1, 0, 0)$
How many copies $l$ of $s \in S$ to add?

Any number $l$ satisfying:

$$\delta_l(s^g) \geq l\tau$$  \hspace{1cm} (1)

$$\delta_s(g + ls) < \tau$$  \hspace{1cm} (2)

$l$ is called a pivot.

A pivot may be found with binary search.

Ineq. (2) ensures that the marginal gain of adding a copy of $s$ later is bounded by $\tau/\gamma d$. 
Threshold Greedy

- How many copies $l$ of $s \in S$ to add?
- Any number $l$ satisfying:

\[ \delta_{ls}(g) \geq l\tau \quad (1) \]
\[ \delta_s(g + ls) < \tau \quad (2) \]

$l$ is called a *pivot*.
ThresholdGreedy

- How many copies $l$ of $s \in S$ to add?
- Any number $l$ satisfying:

  \[ \delta_{ls}(g) \geq l\tau \]  
  \[ \delta_s(g + ls) < \tau \]

  $l$ is called a pivot.

- A pivot may be found with binary search.
ThresholdGreedy

- How many copies $l$ of $s \in S$ to add?
- Any number $l$ satisfying:

$$\delta_{ls}(g) \geq l\tau \quad (1)$$

$$\delta_s(g + ls) < \tau \quad (2)$$

$l$ is called a *pivot*.

- A pivot may be found with binary search.
- Ineq. (2) ensures that the marginal gain of adding a copy of $s$ later is bounded by

$$\tau/\gamma_d$$
Algorithm 1 ThresholdGreedy

1: **Input:** \( f \in \mathcal{F}_b, k \in \mathbb{N}, \kappa, \varepsilon \in (0, 1) \).
2: **Output:** \( g \in \mathbb{N}^S \)
3: \( g \leftarrow 0, M \leftarrow \max_{s \in S} f(s) \).
4: **for** \( \tau = M; \tau \geq \frac{\kappa \varepsilon^2 M}{k}; \tau \leftarrow \kappa \tau \) **do**
5: **for** \( s \in S \) **do**
6: \( l \leftarrow \text{BinarySearchPivot}(f, g, b, s, k, \tau) \)
7: \( g \leftarrow g + ls \)
8: **if** \( \|g\|_1 = k \) **then**
9: \( \text{return } g \)
10: **return** \( g \)
FastGreedy

- Threshold framework analogous to ThresholdGreedy

\[ 1 - e^{-\beta \gamma s - \eta}, \text{ where } \beta \geq \gamma d. \]

\(^2\) Up to a constant factor, which depends on \(\gamma d\).
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments

\[ 1 - e^{-\beta \gamma s - \eta}, \]  
\[ \beta \geq \gamma \delta. \]

\(^2\)Up to a constant factor, which depends on \( \gamma_d \).
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments
  - Depends on the max marginal gain seen so far

\[ 1 - \frac{e^{-\beta} \gamma}{s} - \eta, \quad \beta \geq \gamma d. \]

\(^2\)Up to a constant factor, which depends on \( \gamma_d \).
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments
  - Depends on the max marginal gain seen so far
  - And a stepsize $\beta$ that adaptively increases when more non-submodularity is encountered

\[ 1 - e^{-\beta \gamma s - \eta}, \] where $\beta \geq \gamma d$.

- The same worst-case query complexity as ThresholdGreedy
- Substantial reduction of the number of queries in practice due to the adaptive stepsize $\beta$

\[ 2 \text{Up to a constant factor, which depends on } \gamma_d. \]
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments
  - Depends on the max marginal gain seen so far
  - And a stepsize $\beta$ that adaptively increases when more non-submodularity is encountered
- Improved theoretical performance ratio

$$1 - e^{-\beta s} - \eta,$$

where $\beta \geq \gamma_d$.  

\[ ^2 \text{Up to a constant factor, which depends on } \gamma_d. \]
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments
  - Depends on the max marginal gain seen so far
  - And a stepsize $\beta$ that adaptively increases when more non-submodularity is encountered
- Improved theoretical performance ratio

$$1 - e^{-\beta \gamma s} - \eta,$$

where $\beta \geq \gamma_d$.
- The same\(^2\) worst-case query complexity as ThresholdGreedy

\(^2\)Up to a constant factor, which depends on $\gamma_d$. 
FastGreedy

- Threshold framework analogous to ThresholdGreedy
- But the threshold does not decrease by set increments
  - Depends on the max marginal gain seen so far
  - And a stepsize $\beta$ that adaptively increases when more non-submodularity is encountered
- Improved theoretical performance ratio

$$1 - e^{-\beta \gamma s} - \eta,$$

where $\beta \geq \gamma_d$.

- The same\(^2\) worst-case query complexity as ThresholdGreedy
- Substantial reduction of the number of queries in practice due to the adaptive stepsize $\beta$

\(^2\)Up to a constant factor, which depends on $\gamma_d$. 
Influence Maximization: A General Framework

- Partial incentives
Influence Maximization: A General Framework

- Partial incentives
  - An incentive increases the probability a user will purchase the product independently (become a seed)
Influence Maximization: A General Framework

- Partial incentives
  - An incentive increases the probability a user will purchase the product independently (become a seed)
  - Increases the susceptibility of the user to the influence of his friends
Influence Maximization: A General Framework

- Partial incentives
  - An incentive increases the probability a user will purchase the product independently (become a seed)
  - Increases the susceptibility of the user to the influence of his friends
- Inherently non-submodular
Influence Maximization: A General Framework

- Partial incentives
  - An incentive increases the probability a user will purchase the product independently (become a seed)
  - Increases the susceptibility of the user to the influence of his friends

- Inherently non-submodular
  - We prove a lower bound on the greedy DR ratios
Experimental Results

Activation $\mathbb{A}(g)$ for the solution returned by each algorithm

(a) GrQc (10 levels)

(b) Facebook (100 levels)

- ThresholdGreedy and FastGreedy Exhibit virtually identical quality of solution with StandardGreedy
Experimental Results

Total function queries on the GrQc and Facebook networks

ThresholdGreedy and FastGreedy query the function much fewer times

FastGreedy queries the function much less than StandardGreedy
Experimental Results

Runtime on the GrQc and Facebook networks with 100 levels

(\textbf{e}) GrQc (10 levels) \hspace{5cm} (\textbf{f}) Facebook (100 levels)

- ThresholdGreedy and FastGreedy have a dramatic runtime improvement
- FastGreedy has lower runtime than StandardGreedy
Experimental Results

In overview, our experiments show that are algorithms:

- Exhibit virtually identical quality of solution with StandardGreedy
- Query the function much fewer times
  - Leads to runtime improvement over StandardGreedy
  - FastGreedy further reduces the number of queries while sacrificing little in solution quality
- Implementation at: https://gitlab.com/emallson/lace
- Poster: 145
- Thanks! Questions?