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

Integer Lattice

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

Example

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

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

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

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
  - **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] for all $\mathbf{v} \leq \mathbf{w}$ (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 $f(\mathbf{0}) = 0$.
- Then determine

$$\max_{\|\mathbf{w}\|_1 \leq k} f(\mathbf{w}), \tag{MCC}$$

where $\mathbf{w} = (\mathbf{w}_s)_{s \in S} \in \mathbb{N}^S$, $\|\mathbf{w}\|_1 = \sum_{s \in S} |\mathbf{w}_s|$, $k \in \mathbb{N}$.

[1] for all $\mathbf{v} \leq \mathbf{w}$ (coordinate-wise), $f(\mathbf{v}) \leq f(\mathbf{w})$

Measures of submodularity

- **diminishing-return (DR) ratio** γ_d

Measures of submodularity

- **diminishing-return (DR) ratio** γ_d
  - maximum value such that

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

where $\mathbf{v} \leq \mathbf{w}$ and $\mathbf{s}$ is step in sth direction

Measures of submodularity

- **diminishing-return (DR) ratio** γ_d
  - maximum value such that

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

    where $\mathbf{v} \leq \mathbf{w}$ and $\mathbf{s}$ is step in sth direction
- **weak DR ratio** γ_s

Measures of submodularity

- **diminishing-return (DR) ratio** γ_d
  - maximum value such that

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

  where $\mathbf{v} \leq \mathbf{w}$ and $\mathbf{s}$ is step in sth direction

- **weak DR ratio** γ_s
  - maximum value such that for all $\mathbf{v}, \mathbf{w}$, such that $\mathbf{v} \leq \mathbf{w}$,

$$\gamma_s(f)(f(\mathbf{w}) - f(\mathbf{v})) \leq \sum_{s \in \{\mathbf{w} - \mathbf{v}\}} f(\mathbf{v} + \mathbf{s}) - f(\mathbf{v}).$$

Measures of submodularity

- **diminishing-return (DR) ratio** γ_d
  - maximum value such that

$$\gamma_d\,(f(\mathbf{w}+\mathbf{s})-f(\mathbf{w})) \leq f(\mathbf{v}+\mathbf{s})-f(\mathbf{v}),$$

  where $\mathbf{v} \leq \mathbf{w}$ and $\mathbf{s}$ is step in sth direction
- **weak DR ratio** γ_s
  - maximum value such that for all $\mathbf{v}, \mathbf{w}$, such that $\mathbf{v} \leq \mathbf{w}$,

$$\gamma_s(f)(f(\mathbf{w})-f(\mathbf{v})) \leq \sum_{s \in \{\mathbf{w}-\mathbf{v}\}} f(\mathbf{v}+\mathbf{s})-f(\mathbf{v}).$$

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

Our contributions

- ThresholdGreedy

Our contributions

- ThresholdGreedy
  - Approximation ratio $(1 - e^{-\gamma_d \gamma_s} - \eta)$ for any $\eta > 0$

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

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 $(1 - e^{-\beta \gamma_s} - \eta)$

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 $(1 - e^{-\beta \gamma_s} - \eta)$
  - β is at least γ_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 $(1 - e^{-\beta \gamma_s} - \eta)$
  - β is at least γ_d and is computed by the algorithm
  - Uses non-submodularity to decrease its runtime in practice

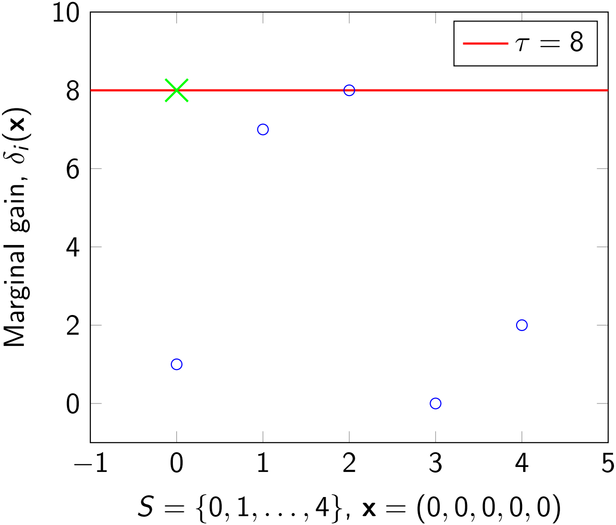
ThresholdGreedy

- Operates by considering decreasing thresholds for the marginal gain

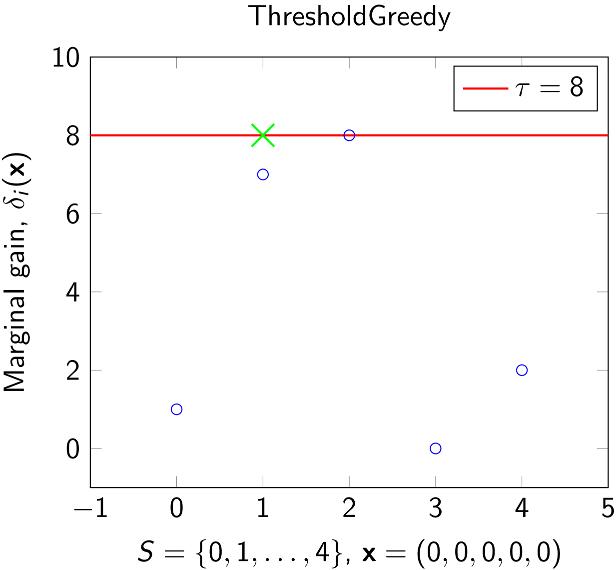
ThresholdGreedy

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

ThresholdGreedy

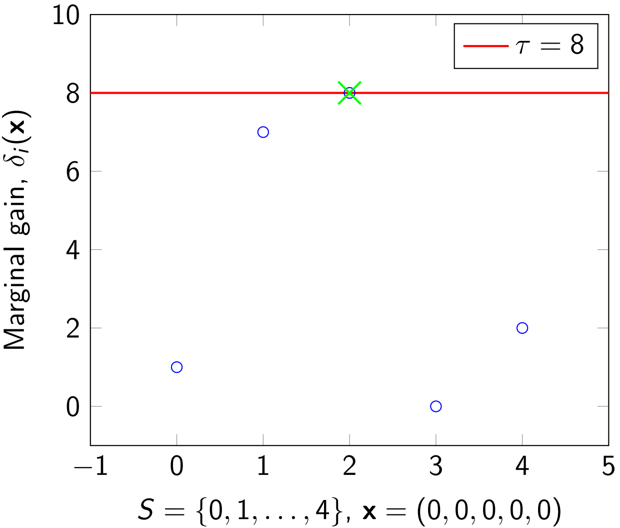


$\delta_0(\mathbf{x}) \geq \tau$?



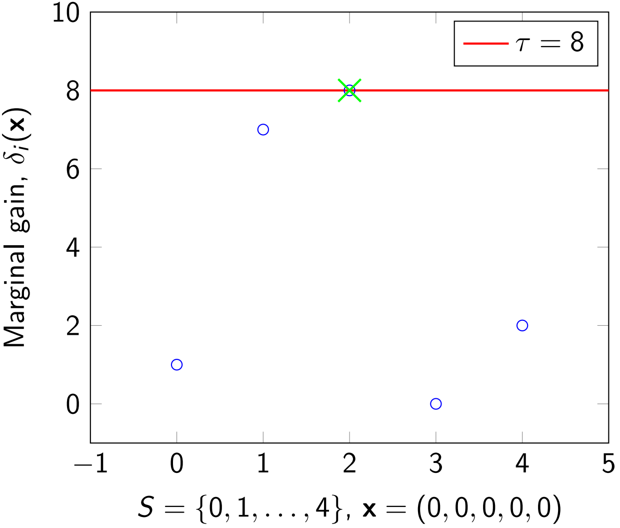
$\delta_1(\mathbf{x}) \geq \tau$?

ThresholdGreedy

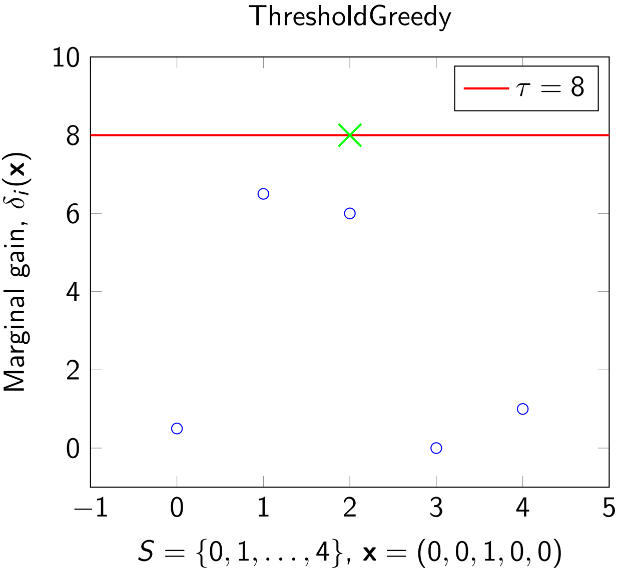


$\delta_2(\mathbf{x}) \geq \tau$?

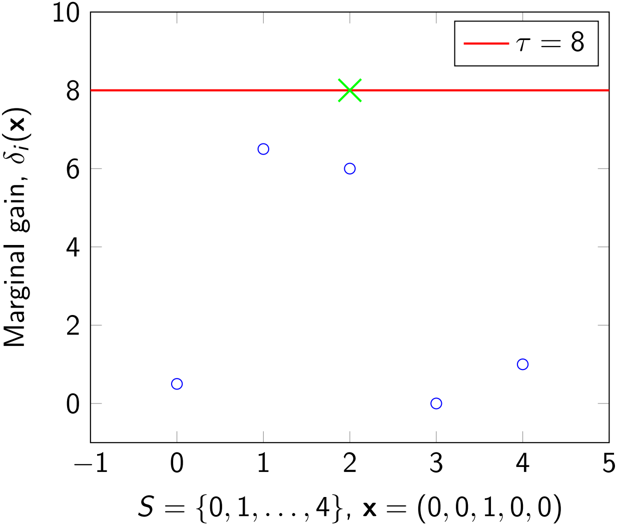
ThresholdGreedy



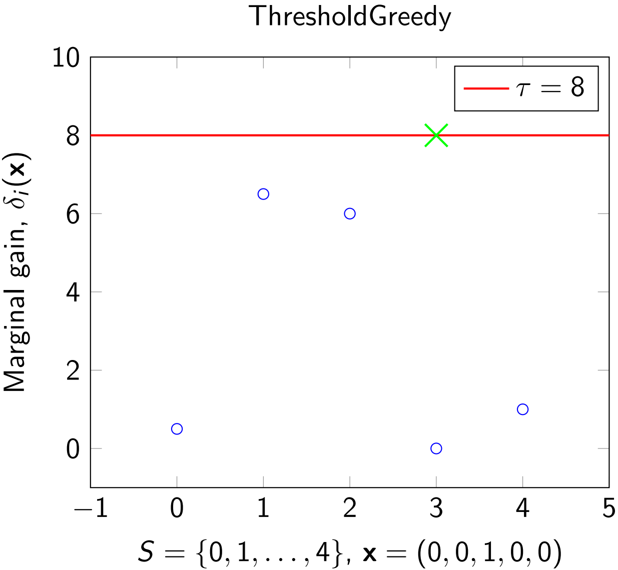
Yes, add it to $\mathbf{x}$


ThresholdGreedy
$\tau = 8$
Marginal gain, $\delta_i(\mathbf{x})$
$S = \{0, 1, \ldots, 4\}$, $\mathbf{x} = (0, 0, 1, 0, 0)$
10
8
6
4
2
0
−1
0
1
2
3
4
5

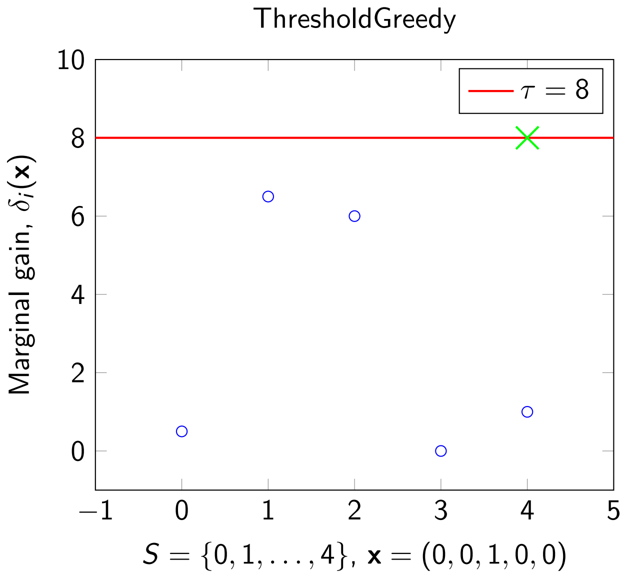
ThresholdGreedy



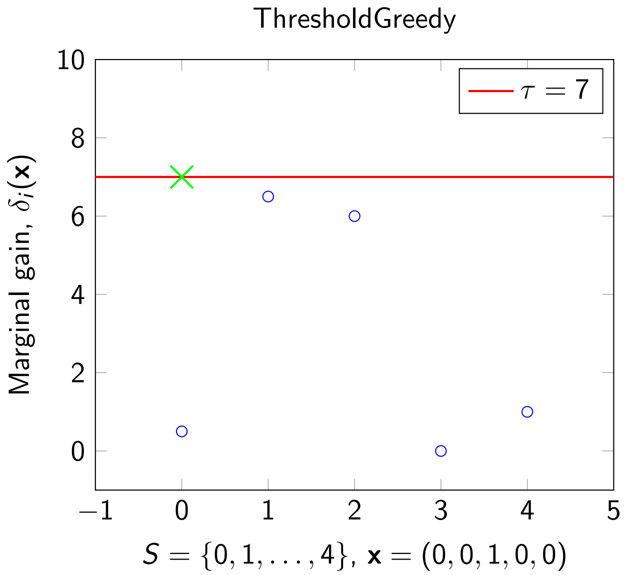
If DR submodular, marginal gains decrease.



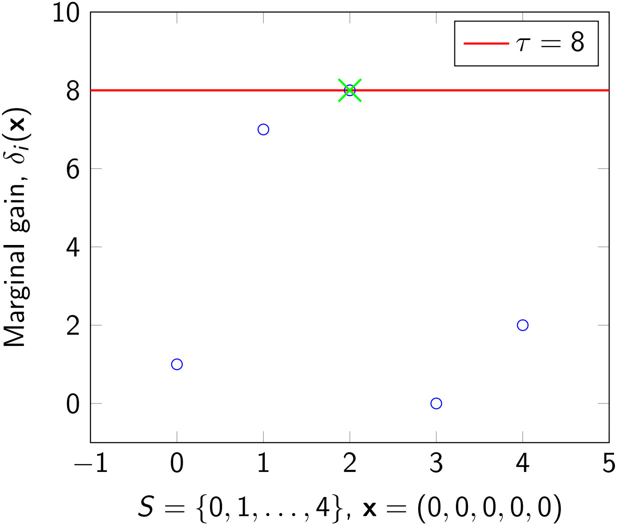
ThresholdGreedy
$\tau = 8$
Marginal gain, $\delta_i(\mathbf{x})$
10
8
6
4
2
0
−1
0
1
2
3
4
5
$S = \{0, 1, \ldots, 4\}$, $\mathbf{x} = (0, 0, 1, 0, 0)$



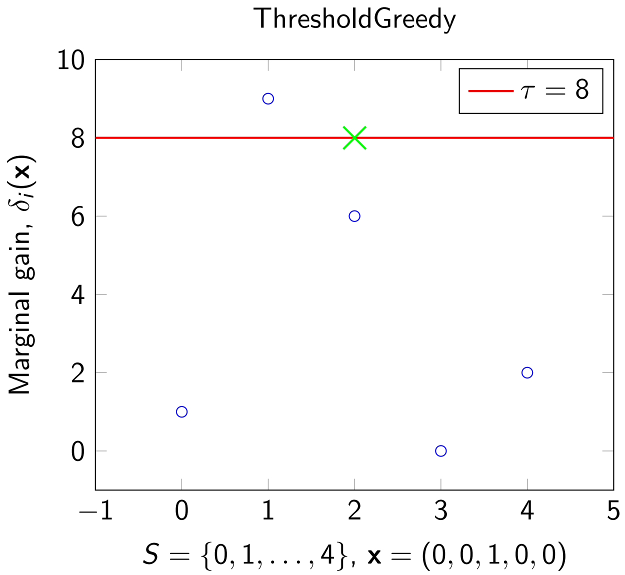
ThresholdGreedy
$\tau = 8$
Marginal gain, $\delta_i(\mathbf{x})$
10
8
6
4
2
0
−1
0
1
2
3
4
5
$S = \{0, 1, \ldots, 4\}$, $\mathbf{x} = (0, 0, 1, 0, 0)$


ThresholdGreedy
$\tau = 7$
Marginal gain, $\delta_i(\mathbf{x})$
10
8
6
4
2
0
−1
0
1
2
3
4
5
$S = \{0, 1, \ldots, 4\}$, $\mathbf{x} = (0, 0, 1, 0, 0)$

ThresholdGreedy



If not submodular, marginal gains may increase. . .



ThresholdGreedy
$\tau = 8$
Marginal gain, $\delta_i(\mathbf{x})$
$S = \{0, 1, \ldots, 4\}$, $\mathbf{x} = (0, 0, 1, 0, 0)$

ThresholdGreedy

- How many copies l of $s \in S$ to add?

ThresholdGreedy

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

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

$$\delta_{\mathbf{s}}(\mathbf{g} + l\mathbf{s}) < \tau \quad (2)$$

l is called a *pivot*.

ThresholdGreedy

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

$$\delta_{l\mathbf{s}}(\mathbf{g}) \geq l\tau \tag{1}$$

$$\delta_{\mathbf{s}}(\mathbf{g} + l\mathbf{s}) < \tau \tag{2}$$

  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:

$$\delta_{l\mathbf{s}}(\mathbf{g}) \geq l\tau \tag{1}$$

$$\delta_{\mathbf{s}}(\mathbf{g} + l\mathbf{s}) < \tau \tag{2}$$

  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

$$\tau / \gamma_d$$

  .

ThresholdGreedy

Algorithm 1 ThresholdGreedy

Input: $f \in \mathcal{F}_{\mathbf{b}}$, $k \in \mathbb{N}$, $\kappa, \varepsilon \in (0, 1)$.
Output: $\mathbf{g} \in \mathbb{N}^S$
$\mathbf{g} \leftarrow \mathbf{0}$, $M \leftarrow \max_{s \in S} f(\mathbf{s})$.
for $\left(\tau = M; \tau \geq \frac{\kappa \varepsilon^2 M}{k}; \tau \leftarrow \kappa\tau\right)$ **do**
    for $s \in S$ **do**
        $l \leftarrow$BinarySearchPivot$(f, \mathbf{g}, \mathbf{b}, s, k, \tau)$
        $\mathbf{g} \leftarrow \mathbf{g} + l\mathbf{s}$
        if $\|\mathbf{g}\|_1 = k$ **then**
            return g
return g

FastGreedy

- Threshold framework analogous to ThresholdGreedy

[2] Up to a constant factor, which depends on γ_d.

FastGreedy

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

[2] Up to a constant factor, which depends on γ_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

[2] Up to a constant factor, which depends on γ_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 β that adaptively increases when more non-submodularity is encountered

[2] Up to a constant factor, which depends on γ_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 β that adaptively increases when more non-submodularity is encountered
- Improved theoretical performance ratio

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

where $\beta \geq \gamma_d$.

[2] Up to a constant factor, which depends on γ_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 β 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 γ_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 β 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 β

[2] Up to a constant factor, which depends on γ_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

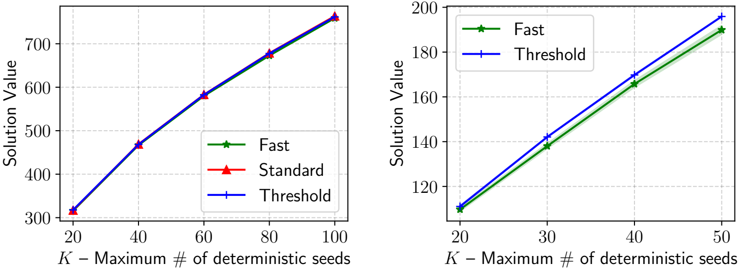
- Partial incentives
  - An incentive increases the probability a user will purchase the product independently (become a seed)
  - Increases the susceptibility of the user to the influence of his friends
- Inherently non-submodular

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

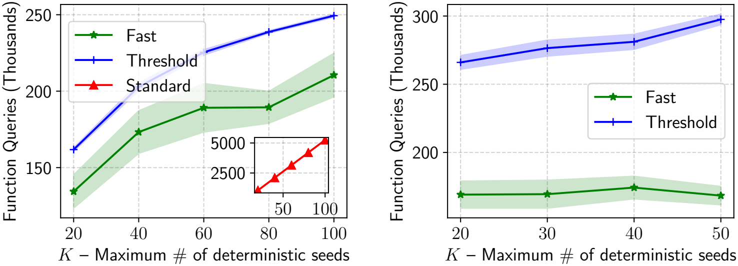


(a) GrQc (10 levels) (b) Facebook (100 levels)

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

Experimental Results

Total function queries on the GrQc and Facebook networks

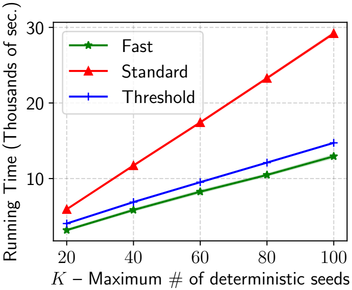


(c) GrQc (10 levels) (d) Facebook (100 levels)

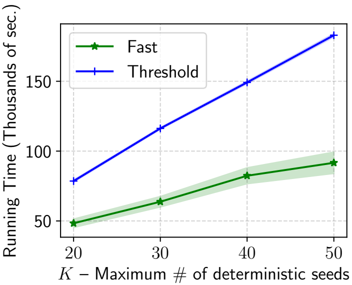
- ThresholdGreedy and FastGreedy query the function much fewer times
- FastGreedy queries the function much less than StandardGreedy

Experimental Results

Runtime on the GrQc and Facebook networks with 100 levels



(e) GrQc (10 levels)



(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?