Optimization of Submodular Functions Tutorial - lecture II

Jan Vondrák[1]

[1] IBM Almaden Research Center
San Jose, CA

Outline

Lecture I:

1. Submodular functions: what and why?
2. Convex aspects: Submodular minimization
3. Concave aspects: Submodular maximization

Lecture II:

1. Hardness of constrained submodular minimization
2. Unconstrained submodular maximization
3. Hardness more generally: the symmetry gap

Hardness of constrained submodular minimization

We saw:

- Submodular minimization is in *P*
  (without constraints, and also under "parity type" constraints).

Hardness of constrained submodular minimization

We saw:

- Submodular minimization is in P
  (without constraints, and also under "parity type" constraints).

However: minimization is brittle and can become very hard to approximate under simple constraints.

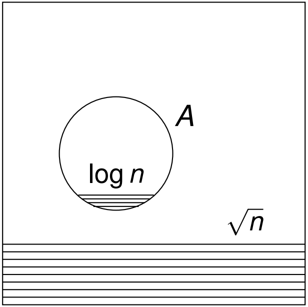
- $\sqrt{\frac{n}{\log n}}$-hardness for $\min\{f(S) : |S| \geq k\}$, Submodular Load Balancing, Submodular Sparsest Cut [Svitkina,Fleischer '09]
- $n^{\Omega(1)}$-hardness for Submodular Spanning Tree, Submodular Perfect Matching, Submodular Shortest Path [Goel,Karande,Tripathi,Wang '09]

These hardness results assume the value oracle model: the only access to f is through value queries, $f(S) = ?$

Superconstant hardness for submodular minimization

Problem: $\min\{f(S) : |S| \geq k\}$.

Construction of [Goemans,Harvey,Iwata,Mirrokni '09]:



A = random (hidden) set of size $k = \sqrt{n}$

$f(S) = \min\{\sqrt{n}, |S \setminus A| + \min\{\log n, |S \cap A|\}\}$

Analysis: with high probability, a value query does not give any information about $A \Rightarrow$ an algorithm will return a set of value $\sqrt{n}$, while the optimum is $\log n$.

Overview of submodular minimization

CONSTRAINED SUBMODULAR MINIMIZATION

Constraint	Approximation	Hardness	hardness ref
Vertex cover	2	2 [UGC]	Khot,Regev ’03
k-unif. hitting set	k	k [UGC]	Khot,Regev ’03
k-way partition	$2 - 2/k$	$2 - 2/k$	Ene,V.,Wu ’12
Facility location	$\log n$	$\log n$	Svitkina,Tardos ’07
Set cover	n	$n/\log^2 n$	Iwata,Nagano ’09
$\lvert S\rvert \geq k$	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer ’09
Sparsest Cut	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer ’09
Load Balancing	$\tilde{O}(\sqrt{n})$	$\tilde{\Omega}(\sqrt{n})$	Svitkina,Fleischer ’09
Shortest path	$O(n^{2/3})$	$\Omega(n^{2/3})$	GKTW ’09
Spanning tree	$O(n)$	$\Omega(n)$	GKTW ’09

Outline

Lecture I:

1. Submodular functions: what and why?
2. Convex aspects: Submodular minimization
3. Concave aspects: Submodular maximization

Lecture II:

1. Hardness of constrained submodular minimization
2. Unconstrained submodular maximization
3. Hardness more generally: the symmetry gap

Maximization of a nonnegative submodular function

We saw:

- Maximizing a submodular function is NP-hard (Max Cut).

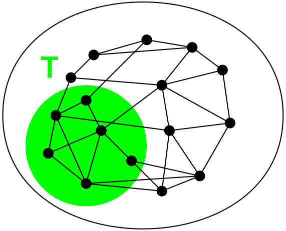
Maximization of a nonnegative submodular function

We saw:

- Maximizing a submodular function is NP-hard (Max Cut).

Unconstrained submodular maximization: *Given a submodular function $f : 2^N \to \mathbb{R}_+$, how well can we approximate the maximum?*

Special case - Max Cut:



polynomial-time 0.878-approximation [Goemans-Williamson '95], best possible assuming the Unique Games Conjecture [Khot,Kindler, Mossel,O'Donnell '04, Mossel,O'Donnell,Oleszkiewicz '05]

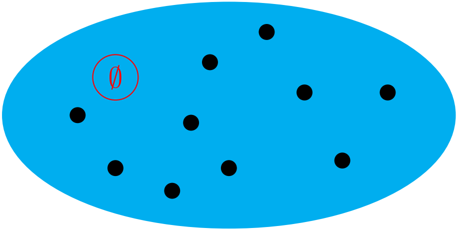
Optimal approximation for submodular maximization

Unconstrained submodular maximization: $\max_{S \subseteq N} f(S)$
has been resolved recently:

- there is a (randomized) $1/2$-approximation [Buchbinder,Feldman,Naor,Schwartz '12]
- $(1/2 + \epsilon)$-approximation in the value oracle model would require exponentially many queries [Feige,Mirrokni,V. '07]
- $(1/2 + \epsilon)$-approximation for certain explicitly represented submodular functions would imply $NP = RP$ [Dobzinski,V. '12]

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

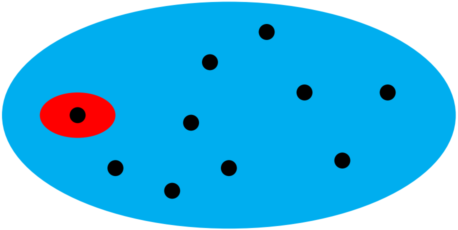


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
 Pick $i \in B \setminus A$;
 Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
 With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
 With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

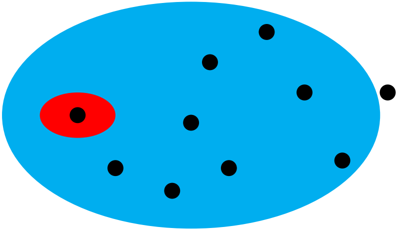


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

```
While A ≠ B {
 Pick i ∈ B \ A;
 Let α = max{f(A + i) − f(A), 0}, β = max{f(B − i) − f(B), 0};
 With probability α/(α+β), include i in A;
 With probability β/(α+β) remove i from B; }
```

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

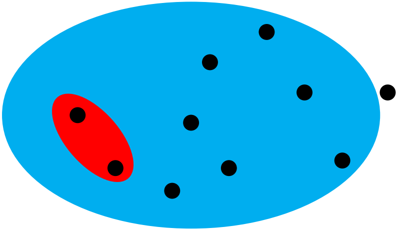


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

```
While A ≠ B {
 Pick i ∈ B \ A;
 Let α = max{f(A + i) − f(A), 0}, β = max{f(B − i) − f(B), 0};
 With probability α/(α+β), include i in A;
 With probability β/(α+β) remove i from B; }
```

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

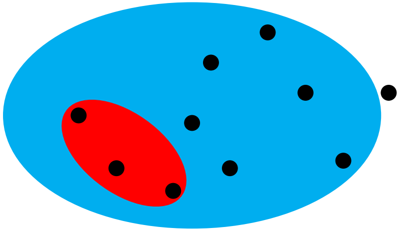


Initialize $A = \emptyset$, $B =$ everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
 Pick $i \in B \setminus A$;
 Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
 With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
 With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

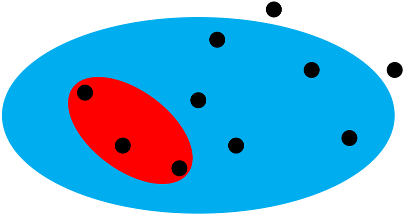


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
Pick $i \in B \setminus A$;
Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

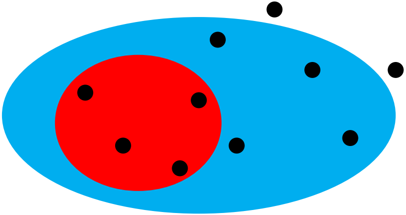


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
Pick $i \in B \setminus A$;
Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

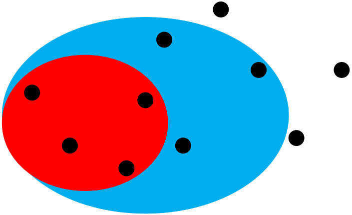


Initialize $A = \emptyset$, $B =$ everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
 Pick $i \in B \setminus A$;
 Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
 With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
 With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

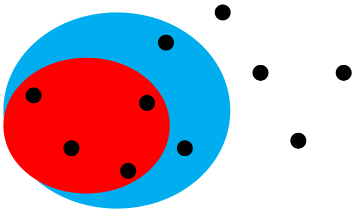


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
Pick $i \in B \setminus A$;
Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

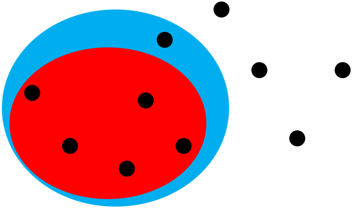


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
Pick $i \in B \setminus A$;
Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

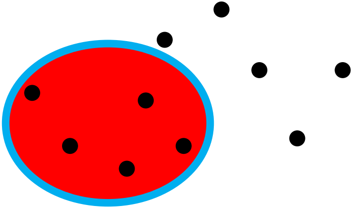


Initialize $A = \emptyset$, $B =$everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
Pick $i \in B \setminus A$;
Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

$\frac{1}{2}$-approximation for submodular maximization [Buchbinder,Feldman,Naor,Schwartz '12]

A double-greedy algorithm with two evolving solutions:

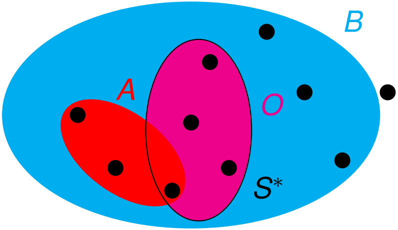


Initialize $A = \emptyset$, $B =$ everything.
In each step, grow A or shrink B.
Invariant: $A \subseteq B$.

While $A \neq B$ {
 Pick $i \in B \setminus A$;
 Let $\alpha = \max\{f(A+i) - f(A), 0\}$, $\beta = \max\{f(B-i) - f(B), 0\}$;
 With probability $\frac{\alpha}{\alpha+\beta}$, include i in A;
 With probability $\frac{\beta}{\alpha+\beta}$ remove i from B; }

Analysis of $\frac{1}{2}$-approximation

Evolving optimum: $O = A \cup (B \cap S^*)$, where S^* is the optimum.
We track the quantity $f(A) + f(B) + 2f(O)$:

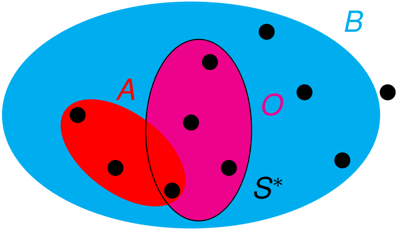


Initially: $A = \emptyset$, $B = N$, $O = S^*$.
$f(A) + f(B) + 2f(O) \geq 2 \cdot OPT$.

At the end: $A = B = O =$ output.
$f(A) + f(B) + 2f(O) = 4 \cdot ALG$.

Analysis of $\frac{1}{2}$-approximation

Evolving optimum: $O = A \cup (B \cap S^*)$, where S^* is the optimum.
We track the quantity $f(A) + f(B) + 2f(O)$:



Initially: $A = \emptyset$, $B = N$, $O = S^*$.
$f(A) + f(B) + 2f(O) \geq 2 \cdot OPT$.

At the end: $A = B = O =$ output.
$f(A) + f(B) + 2f(O) = 4 \cdot ALG$.

Claim: $\mathbb{E}[f(A) + f(B) + 2f(O)]$ never decreases in the process.
Proof: Expected change in $f(A) + f(B) + 2f(O)$ is

$$\frac{\alpha}{\alpha+\beta} \cdot \alpha + \frac{\beta}{\alpha+\beta} \cdot \beta - \frac{2\alpha\beta}{\alpha+\beta} = \frac{(\alpha-\beta)^2}{\alpha+\beta} \geq 0.$$

Optimality of $1/2$ for submodular maximization

How do we prove that $1/2$ *is optimal?* [Feige, Mirrokni, V. '07]

Optimality of $1/2$ for submodular maximization

How do we prove that $1/2$ *is optimal?* [Feige, Mirrokni, V. '07]

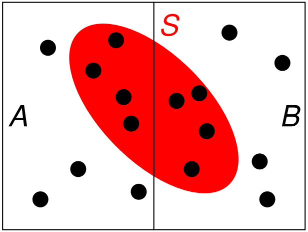
Again, the value oracle model: the only access to f is through value queries, $f(S) = ?$, polynomially many times.

Optimality of $1/2$ for submodular maximization

How do we prove that $1/2$ *is optimal?* [Feige, Mirrokni, V. '07]

Again, the value oracle model: the only access to f is through value queries, $f(S) = ?$, polynomially many times.

Idea: Construct an instance of optimum $f(S^*) = 1 - \epsilon$, so that all the sets an algorithm will ever see have value $f(S) \leq 1/2$.



$$f(S) = \psi\left(\frac{|S \cap A|}{|A|}, \frac{|S \cap B|}{|B|}\right)$$

A, B are the intended optimal solutions, but the partition (A, B) is *hard to find*.

Constructing the hard instance

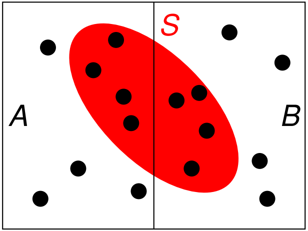
Continuous submodularity:
If $\frac{\partial^2 \psi}{\partial x \partial y} \leq 0$, *then* $f(S) = \psi(\frac{|S \cap A|}{|A|}, \frac{|S \cap B|}{|B|})$ *is submodular.*
(non-increasing partial derivatives $\simeq$ non-increasing marginal values)

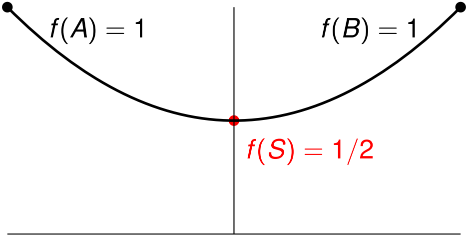
Constructing the hard instance

Continuous submodularity:

If $\frac{\partial^2\psi}{\partial x \partial y} \leq 0$, *then* $f(S) = \psi(\frac{|S \cap A|}{|A|}, \frac{|S \cap B|}{|B|})$ *is submodular.*
(non-increasing partial derivatives $\simeq$ non-increasing marginal values)

The function will be "roughly": $\psi(x, y) = x(1 - y) + (1 - x)y$.

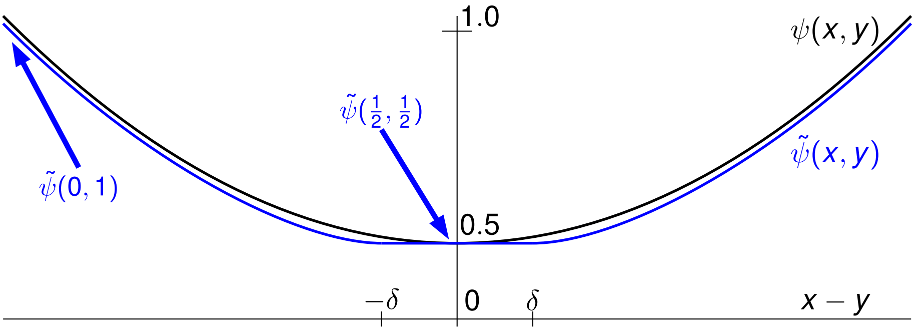




However, it should be hard to find the partition (A, B)*!*

The perturbation trick

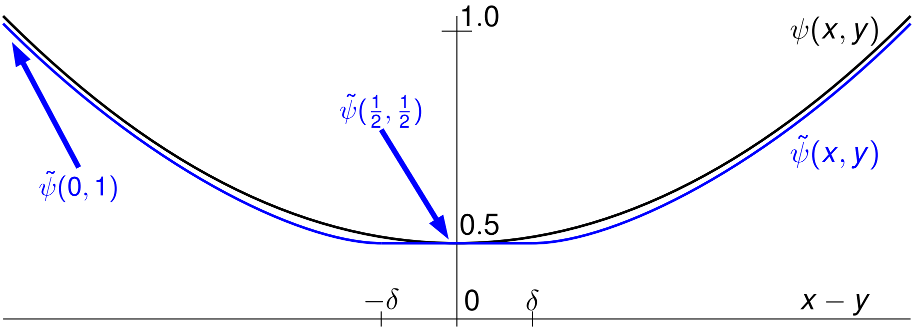
We modify $\psi(x, y)$ as follows:
(graph restricted to $x + y = 1$)



- The function for $|x - y| < \delta$ is flattened so it depends only on $x + y$.

The perturbation trick

We modify $\psi(x, y)$ as follows:
(graph restricted to $x + y = 1$)



- The function for $|x - y| < \delta$ is flattened so it depends only on $x + y$.
- If the partition (A, B) is random, $x = \frac{|S \cap A|}{|A|}$ and $y = \frac{|S \cap B|}{|B|}$ are random variables, with high probability satisfying $|x - y| < \delta$.
- Hence, an algorithm will never learn any information about (A, B).

Hardness and symmetry

Conclusion: for unconstrained submodular maximization,

- The optimum is $f(A) = f(B) = 1 - \epsilon$.
- An algorithm can only find solutions symmetrically split between A, B: $|S \cap A| \simeq |S \cap B|$.
- The value of such solutions is at most $1/2$.

Hardness and symmetry

Conclusion: for unconstrained submodular maximization,

- The optimum is $f(A) = f(B) = 1 - \epsilon$.
- An algorithm can only find solutions symmetrically split between A, B: $|S \cap A| \simeq |S \cap B|$.
- The value of such solutions is at most $1/2$.

More general view:

- The difficulty here is in distinguishing between symmetric and asymmetric solutions.
- Submodularity is flexible enough that we can hide the asymmetric solutions and force an algorithm to find only symmetric ones.

Outline

Lecture I:

1. Submodular functions: what and why?
2. Convex aspects: Submodular minimization
3. Concave aspects: Submodular maximization

Lecture II:

1. Hardness of constrained submodular minimization
2. Unconstrained submodular maximization
3. Hardness more generally: the symmetry gap

Symmetric instances

Symmetric instance: $\max\{f(S) : S \in \mathcal{F}\}$ on a ground set X is symmetric under a group of permutations $\mathcal{G} \subset \mathbb{S}(X)$, if for any $\sigma \in \mathcal{G}$,

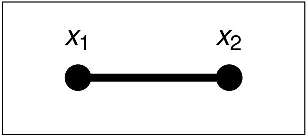
- $f(S) = f(\sigma(S))$
- $S \in \mathcal{F} \Leftrightarrow S' \in \mathcal{F}$ whenever $\overline{\mathbf{1}_S} = \overline{\mathbf{1}_{S'}}$, where
- $\bar{x} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$ *(symmetrization operation)*

Symmetric instances

Symmetric instance: $\max\{f(S) : S \in \mathcal{F}\}$ on a ground set X is symmetric under a group of permutations $\mathcal{G} \subset \mathbb{S}(X)$, if for any $\sigma \in \mathcal{G}$,

- $f(S) = f(\sigma(S))$
- $S \in \mathcal{F} \Leftrightarrow S' \in \mathcal{F}$ whenever $\overline{\mathbf{1}_S} = \overline{\mathbf{1}_{S'}}$, where
- $\bar{x} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(x)]$ *(symmetrization operation)*

Example: Max Cut on K_2



- $X = \{1, 2\}$, $\mathcal{F} = 2^X$, $P(\mathcal{F}) = [0, 1]^2$.
- $f(S) = 1$ if $|S| = 1$, otherwise 0.
- Symmetric under $\mathcal{G} = \mathbb{S}_2$, all permutations of 2 elements.
- For $x = (x_1, x_2)$, $\bar{x} = (\frac{x_1+x_2}{2}, \frac{x_1+x_2}{2})$.

Symmetry gap:

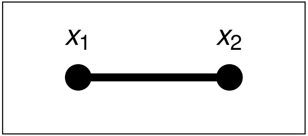
$$\gamma = \frac{\overline{OPT}}{OPT}$$

where

$$OPT = \max\{F(x) : x \in P(\mathcal{F})\}$$
$$\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\}$$

where $F(x)$ is the multilinear extension of f.

Example:



- $OPT = \max\{F(x) : x \in P(\mathcal{F})\} = F(1,0) = 1.$
- $\overline{OPT} = \max\{F(\bar{x}) : x \in P(\mathcal{F})\} = F(\frac{1}{2}, \frac{1}{2}) = 1/2.$

Symmetry gap $\Rightarrow$ hardness

Oracle hardness [V. '09]:
For any instance $\mathcal{I}$ of submodular maximization with symmetry gap γ, and any $\epsilon > 0$, $(\gamma + \epsilon)$-approximation for a class of instances produced by "blowing up" $\mathcal{I}$ would require exponentially many value queries.

Computational hardness [Dobzinski, V. '12]:
There is no $(\gamma + \epsilon)$-approximation for a certain explicit representation of these instances, unless NP $=$ RP.

Symmetry gap $\Rightarrow$ hardness

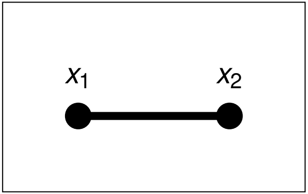
Oracle hardness [V. '09]:
For any instance $\mathcal{I}$ of submodular maximization with symmetry gap γ, and any $\epsilon > 0$, $(\gamma + \epsilon)$-approximation for a class of instances produced by "blowing up" $\mathcal{I}$ would require exponentially many value queries.

Computational hardness [Dobzinski, V. '12]:
There is no $(\gamma + \epsilon)$-approximation for a certain explicit representation of these instances, unless $NP = RP$.

Notes:

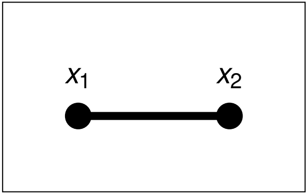
- "Blow-up" means expanding the ground set, replacing the objective function by the perturbed one, and extending the feasibility constraint in a natural way.
- Example: $\max\{f(S) : |S| \leq 1\}$ on a ground set $[k]$ $\longrightarrow \max\{f(S) : |S| \leq n/k\}$ on a ground set $[n]$.

Application 1: nonnegative submodular maximization



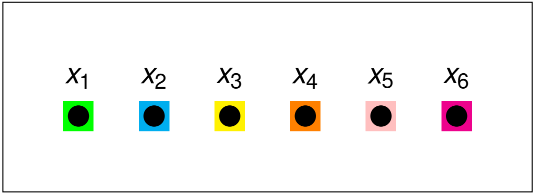
- $\max\{f(S) : S \subseteq \{1,2\}\}$: symmetric under $\mathbb{S}_2$.
- Symmetry gap is $\gamma = 1/2$.
- Refined instances are instances of unconstrained (non-monotone) submodular maximization.

Application 1: nonnegative submodular maximization



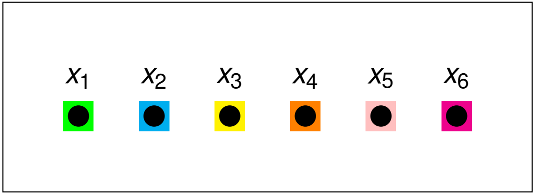
- $\max\{f(S) : S \subseteq \{1,2\}\}$: symmetric under $\mathbb{S}_2$.
- Symmetry gap is $\gamma = 1/2$.
- Refined instances are instances of unconstrained (non-monotone) submodular maximization.
- Theorem implies that a better than $1/2$-approximation is impossible (previously known [FMV '07]).

Application 2: submodular welfare maximization



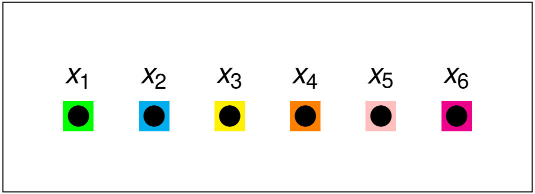
- k items, k players; each player has a valuation function $f(S) = \min\{|S|, 1\}$, symmetric under $\mathbb{S}_k$.

Application 2: submodular welfare maximization



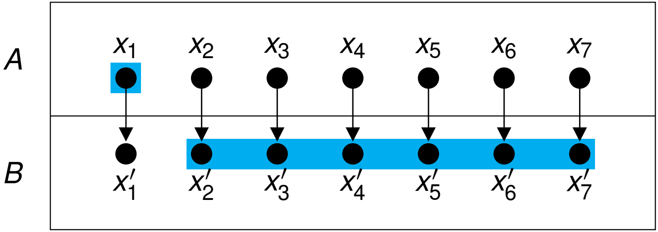
- k items, k players; each player has a valuation function $f(S) = \min\{|S|, 1\}$, symmetric under $\mathbb{S}_k$.
- Optimum allocates 1 item to each player, $OPT = k$.
- $\overline{OPT} = k \cdot F(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}) = k(1 - (1 - \frac{1}{k})^k)$.

Application 2: submodular welfare maximization



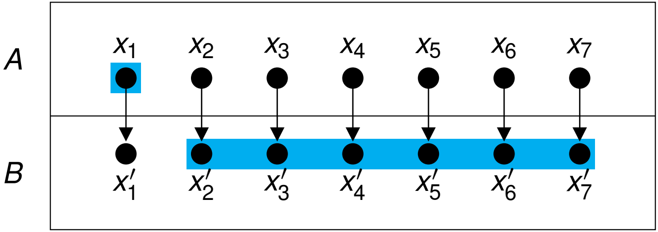
- k items, k players; each player has a valuation function $f(S) = \min\{|S|, 1\}$, symmetric under $\mathbb{S}_k$.
- Optimum allocates 1 item to each player, $OPT = k$.
- $\overline{OPT} = k \cdot F(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}) = k(1 - (1 - \frac{1}{k})^k)$.
- $\Rightarrow$ hardness of $(1 - (1 - 1/k)^k + \epsilon)$-approximation for k players [Mirrokni,Schapira,V. '08]
- $(1 - (1 - 1/k)^k)$-approximation can be achieved [Feldman,Naor,Schwartz '11]

Application 3: non-monotone submodular over bases



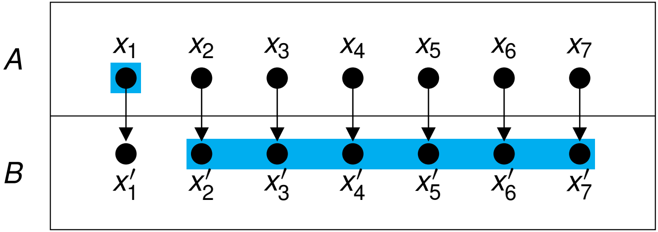
- $X = A \cup B$, $|A| = |B| = k$,
  $\mathcal{F} = \{S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1\}$.
- $f(S) =$ number of arcs leaving S; symmetric under $\mathbb{S}_k$.

Application 3: non-monotone submodular over bases



- $X = A \cup B$, $|A| = |B| = k$,
  $\mathcal{F} = \{S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1\}$.
- $f(S)$ = number of arcs leaving S; symmetric under $\mathbb{S}_k$.
- $OPT = F(1, 0, \ldots, 0; 0, 1, \ldots, 1) = 1$.
- $\overline{OPT} = F(\frac{1}{k}, \ldots, \frac{1}{k}; 1 - \frac{1}{k}, \ldots, 1 - \frac{1}{k}) = \frac{1}{k}$.

Application 3: non-monotone submodular over bases



- $X = A \cup B$, $|A| = |B| = k$,
  $\mathcal{F} = \{S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1\}$.
- $f(S)$ = number of arcs leaving S; symmetric under $\mathbb{S}_k$.
- $OPT = F(1, 0, \ldots, 0; 0, 1, \ldots, 1) = 1$.
- $\overline{OPT} = F(\frac{1}{k}, \ldots, \frac{1}{k}; 1 - \frac{1}{k}, \ldots, 1 - \frac{1}{k}) = \frac{1}{k}$.
- Refined instances: non-monotone submodular maximization over matroid bases, with base packing number $\nu = k/(k-1)$.
- Theorem implies that a better than $\frac{1}{k}$-approximation is impossible.

Symmetry gap ↔ Integrality gap

In fact: [Ene,V.,Wu '12]

- Symmetry gap is equal to the integrality gap of a related LP.
- In some cases, LP gap gives a matching UG-hardness result.

Symmetry gap $\leftrightarrow$ Integrality gap

In fact: [Ene,V.,Wu '12]

- Symmetry gap is equal to the integrality gap of a related LP.
- In some cases, LP gap gives a matching UG-hardness result.

Example: both gaps are $2 - 2/k$ for Node-weighted k-way Cut.

- $\Rightarrow$ No $(2 - 2/k + \epsilon)$-approximation for Node-weighted k-way Cut (assuming UGC).
- $\Rightarrow$ No $(2 - 2/k + \epsilon)$-approximation for Submodular k-way Partition (in the value oracle model)
- $(2 - 2/k)$-approximation can be achieved for both.

Hardness results from symmetry gap (in red)

MONOTONE MAXIMIZATION

Constraint	Approximation	Hardness	hardness ref
$\|S\| \leq k$, matroid	$1 - 1/e$	$1 - 1/e$	Nemhauser,Wolsey '78
k-player welfare	$1 - (1 - \frac{1}{k})^k$	$1 - (1 - \frac{1}{k})^k$	Mirrokni,Schapira,V. '08
k matroids	$k + \epsilon$	$\Omega(k/\log k)$	Hazan,Safra,Schwartz'03

NON-MONOTONE MAXIMIZATION

Constraint	Approximation	Hardness	hardness ref
unconstrained	$1/2$	$1/2$	Feige,Mirrokni,V. '07
$\|S\| \leq k$	$1/e$	0.49	Oveis-Gharan,V. '11
matroid	$1/e$	0.48	Oveis-Gharan,V. '11
matroid base	$\frac{1}{2}(1 - \frac{1}{\nu})$	$1 - \frac{1}{\nu}$	V. '09
k matroids	$k + O(1)$	$\Omega(k/\log k)$	Hazan,Safra,Schwartz '03

Where to go next?

Many questions unanswered: optimal approximations, online algorithms, stochastic models, incentive-compatible mechanisms, more powerful oracle models,...

Two meta-questions:

- Is there a maximization problem which is significantly more difficult for monotone submodular functions than for linear functions?
- Can the symmetry gap ratio be always achieved, for problems where the multilinear relaxation can be rounded without loss?