Optimization of Submodular Functions
Tutorial - lecture II

Jan Vondrák

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
We saw:

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

\[
A = \text{random (hidden) set of size } k = \sqrt{n}
\]

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

**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

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>hardness ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex cover</td>
<td>2</td>
<td>$2_{[UGC]}$</td>
<td>Khot, Regev ’03</td>
</tr>
<tr>
<td>$k$-unif. hitting set</td>
<td>$k$</td>
<td>$k_{[UGC]}$</td>
<td>Khot, Regev ’03</td>
</tr>
<tr>
<td>$k$-way partition</td>
<td>$2 - 2/k$</td>
<td>$2 - 2/k$</td>
<td>Ene, V., Wu ’12</td>
</tr>
<tr>
<td>Facility location</td>
<td>$\log n$</td>
<td>$\log n$</td>
<td>Svitkina, Tardos ’07</td>
</tr>
<tr>
<td>Set cover</td>
<td>$n$</td>
<td>$n/\log^2 n$</td>
<td>Iwata, Nagano ’09</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
<td>\geq k$</td>
<td>$\tilde{O}(\sqrt{n})$</td>
</tr>
<tr>
<td>Sparsest Cut</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$\tilde{\Omega}(\sqrt{n})$</td>
<td>Svitkina, Fleischer ’09</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$\tilde{\Omega}(\sqrt{n})$</td>
<td>Svitkina, Fleischer ’09</td>
</tr>
<tr>
<td>Shortest path</td>
<td>$O(n^{2/3})$</td>
<td>$\Omega(n^{2/3})$</td>
<td>GKTW ’09</td>
</tr>
<tr>
<td>Spanning tree</td>
<td>$O(n)$</td>
<td>$\Omega(n)$</td>
<td>GKTW ’09</td>
</tr>
</tbody>
</table>
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]
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]
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 \( A = \emptyset \), \( B = \text{everything} \).
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 \( B \).}
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 $A = \emptyset$, $B = \text{everything}$.

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 \( A = \emptyset \), \( B = \text{everything} \).
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 \( B \);
}
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 $B$;
}
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 $B$;
}
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 $B$;
}
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 $B$;}

1/2-approximation for submodular maximization 
[Buchbinder,Feldman,Naor,Schwartz ’12]
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: $A = B = O = \text{output}$. 
$f(A) + f(B) + 2f(O) = 4 \cdot ALG$. 

Claim: $E[f(A) + f(B) + 2f(O)]$ never decreases in the process.
Proof: Expected change in $f(A) + f(B) + 2f(O)$ is $\alpha \alpha + \beta \cdot \alpha + \beta \cdot \beta - 2 \alpha \beta \alpha + \beta \geq 0$. 

Jan Vondrák (IBM Almaden)
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: $A = \emptyset$, $B = N$, $O = S^*$. 
$f(A) + f(B) + 2f(O) \geq 2 \cdot \text{OPT}$.

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

**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]
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 $f(S) \leq 1/2$.

![Diagram](image)

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

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

If $\frac{\partial^2 \psi}{\partial x \partial y} \leq 0$, then $f(S) = \psi\left( \frac{|S \cap A|}{|A|}, \frac{|S \cap B|}{|B|} \right)$ is submodular.

(non-increasing partial derivatives $\simeq$ non-increasing marginal values)
Constructing the hard instance

Continuous submodularity:

\[ \frac{\partial^2 \psi}{\partial x \partial y} \leq 0, \text{ then } f(S) = \psi\left( \frac{|S \cap A|}{|A|}, \frac{|S \cap B|}{|B|} \right) \text{ is submodular.} \]

(non-increasing partial derivatives \(\approx\) non-increasing marginal values)

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

\[ f(A) = 1 \]
\[ f(B) = 1 \]
\[ f(S) = 1/2 \]

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 $(A, B)$. 

Jan Vondrák (IBM Almaden)
**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 $1/2$. 
**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 \( G \subset \mathcal{S}(X) \), if for any \( \sigma \in G \),
- \( f(S) = f(\sigma(S)) \)
- \( S \in \mathcal{F} \iff S' \in \mathcal{F} \) whenever \( \bar{1}_S = \bar{1}_{S'} \), where
- \( \bar{x} = \mathbb{E}_{\sigma \in G}[\sigma(x)] \) (*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 \( \mathcal{G} \subset \mathcal{S}(X) \), if for any \( \sigma \in \mathcal{G} \),

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

**Example:** Max Cut on \( K_2 \)

\[
\begin{array}{c}
\text{Example: Max Cut on } K_2 \\
\end{array}
\]

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

**Oracle hardness** [V. ’09]:
For any instance $\mathcal{I}$ of submodular maximization with symmetry gap $\gamma$, 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 $[k]$
  $\quad\rightarrow$ max\{$f(S) : |S| \leq n/k$\} on a ground set $[n]$.
Application 1: nonnegative submodular maximization

\[
\max \{ f(S) : S \subseteq \{1, 2\} \}: \text{symmetric under } S_2.
\]

- Symmetry gap is \( \gamma = 1/2 \).
- Refined instances are instances of unconstrained (non-monotone) submodular maximization.
Application 1: nonnegative submodular maximization

\[ \max \{ f(S) : S \subseteq \{1, 2\} \} : \text{symmetric under } S_2. \]

- 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 \( S_k \).
Application 2: submodular welfare maximization

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

Optimum allocates 1 item to each player, $OPT = k$.

$OPT = k \cdot F(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}) = k(1 - (1 - \frac{1}{k})^k)$.

$\Rightarrow$ 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

\[ X = A \cup B, \ |A| = |B| = k, \]
\[ \mathcal{F} = \{ S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1 \}. \]
\[ f(S) = \text{number of arcs leaving } S; \text{ symmetric under } S_k. \]
### Application 3: non-monotone submodular over bases

Given $X = A \cup B$, $|A| = |B| = k$, $\mathcal{F} = \{ S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1 \}$.

- $f(S) =$ number of arcs leaving $S$; symmetric under $S_k$.
- $OPT = F(1, 0, \ldots, 0; 0, 1, \ldots, 1) = 1$.
- $\overline{OPT} = F(\frac{1}{k}, \ldots, \frac{1}{k}; 1 - \frac{1}{k}, \ldots, 1 - \frac{1}{k}) = \frac{1}{k}$.

**Diagram:**

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x'_1$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$x'_2$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$x'_3$</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$x'_4$</td>
</tr>
<tr>
<td>$x_5$</td>
<td>$x'_5$</td>
</tr>
<tr>
<td>$x_6$</td>
<td>$x'_6$</td>
</tr>
<tr>
<td>$x_7$</td>
<td>$x'_7$</td>
</tr>
</tbody>
</table>

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

Refined instances: non-monotone submodular maximization over matroid bases, with base packing number $\nu = k / (k - 1)$.

Theorem implies that a better than $1/k$-approximation is impossible.
Application 3: non-monotone submodular over bases

\[ X = A \cup B, \ |A| = |B| = k, \]
\[ \mathcal{F} = \{ S \subseteq X : |S \cap A| = 1, |S \cap B| = k - 1 \}. \]
\[ f(S) = \text{number of arcs leaving } S; \text{ symmetric under } S_k. \]
\[ OPT = F(1, 0, \ldots, 0; 0, 1, \ldots, 1) = 1. \]
\[ OPT = F\left(\frac{1}{k}, \ldots, \frac{1}{k}; 1 - \frac{1}{k}, \ldots, 1 - \frac{1}{k}\right) = \frac{1}{k}. \]

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 $\leftrightarrow$ 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 $2 - \frac{2}{k}$ for Node-weighted $k$-way Cut.

$\Rightarrow$ No $(2 - \frac{2}{k} + \epsilon)$-approximation for Node-weighted $k$-way Cut (assuming UGC).

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

- $\Rightarrow$ No $(2 - \frac{2}{k} + \epsilon)$-approximation for Node-weighted $k$-way Cut (assuming UGC).
- $\Rightarrow$ No $(2 - \frac{2}{k} + \epsilon)$-approximation for Submodular $k$-way Partition (in the value oracle model)
- $(2 - \frac{2}{k})$-approximation can be achieved for both.
### HARDNESS RESULTS FROM SYMMETRY GAP (IN RED)

**MONOTONE MAXIMIZATION**

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>Hardness ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>S</td>
<td>\leq k$, matroid</td>
<td>$1 - 1/e$</td>
</tr>
<tr>
<td>$k$-player welfare</td>
<td>$1 - (1 - \frac{1}{k})^k$</td>
<td>$1 - (1 - \frac{1}{k})^k$</td>
<td>Mirrokni, Schapira, V. ’08</td>
</tr>
<tr>
<td>$k$ matroids</td>
<td>$k + \epsilon$</td>
<td>$\Omega(k/\log k)$</td>
<td>Hazan, Safra, Schwartz’03</td>
</tr>
</tbody>
</table>

**NON-MONOTONE MAXIMIZATION**

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Approximation</th>
<th>Hardness</th>
<th>Hardness ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>unconstrained</td>
<td>$1/2$</td>
<td>$1/2$</td>
<td>Feige, Mirrokni, V. ’07</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
<td>\leq k$</td>
<td>$1/e$</td>
</tr>
<tr>
<td>matroid</td>
<td>$1/e$</td>
<td>$0.48$</td>
<td>Oveis-Gharan, V. ’11</td>
</tr>
<tr>
<td>matroid base</td>
<td>$\frac{1}{2} (1 - \frac{1}{\nu})$</td>
<td>$1 - \frac{1}{\nu}$</td>
<td>V. ’09</td>
</tr>
<tr>
<td>$k$ matroids</td>
<td>$k + O(1)$</td>
<td>$\Omega(k/\log k)$</td>
<td>Hazan, Safra, Schwartz ’03</td>
</tr>
</tbody>
</table>
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?