Submodular Functions and Their Applications

Jan Vondrák[1]

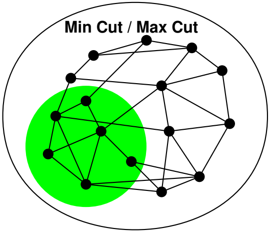
[1] IBM Almaden Research Center
San Jose, CA

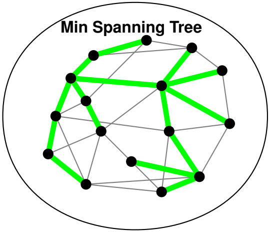
SIAM Discrete Math conference, Minneapolis, MN
June 2014

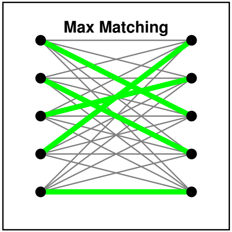
Discrete optimization

What is a discrete optimization problem?

- Find a solution S in a *finite set* of feasible solutions $\mathcal{F} \subset \{0,1\}^n$
- Maximize/minimize an objective function $f(S)$







Some problems are in P:

Min Spanning Tree, Max Flow, Min Cut, Max Matching,...

Many problems are NP-hard:

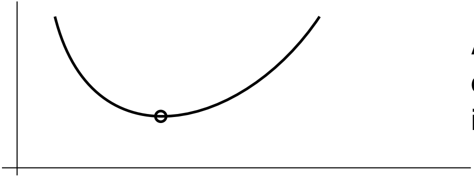
Traveling Salesman, Max Clique, Max Cut, Set Cover, Knapsack,...

Continuous optimization

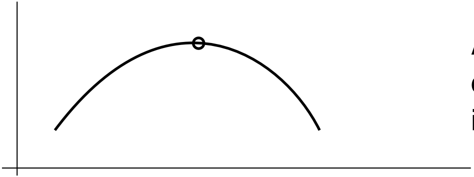
What makes *continuous optimization* tractable?

Continuous optimization

What makes *continuous optimization* tractable?



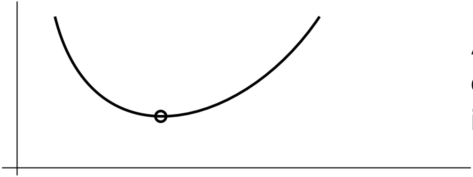
A function $f : \mathbb{R}^n \to \mathbb{R}$
can be minimized efficiently,
if it is convex.



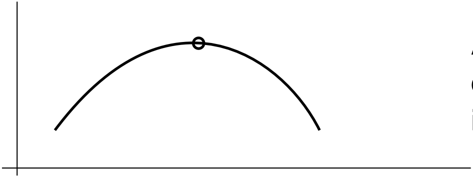
A function $f : \mathbb{R}^n \to \mathbb{R}$
can be maximized efficiently,
if it is concave.

Continuous optimization

What makes *continuous optimization* tractable?



A function $f : \mathbb{R}^n \to \mathbb{R}$
can be minimized efficiently,
if it is convex.



A function $f : \mathbb{R}^n \to \mathbb{R}$
can be maximized efficiently,
if it is concave.

Discrete analogy?
Not so obvious... f is now a set function, or equivalently

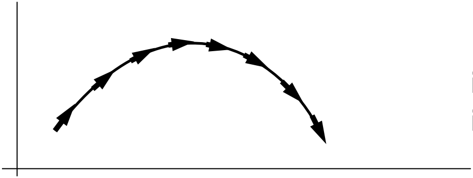
$$f : \{0,1\}^n \to \mathbb{R}.$$

Outline

1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Continuous relaxations for submodular optimization problems.
4. Hardness from symmetric instances.
5. Where do we go next...

From concavity to submodularity

Concavity:

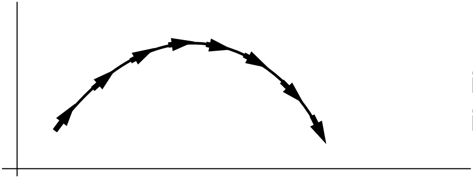


$f : \mathbb{R} \to \mathbb{R}$ is concave,

if the derivative $f'(x)$ is non-increasing in x.

From concavity to submodularity

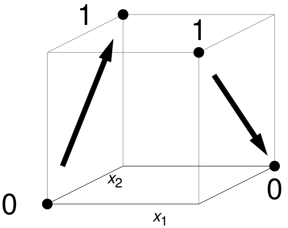
Concavity:



$f : \mathbb{R} \to \mathbb{R}$ is concave,

if the derivative $f'(x)$ is non-increasing in x.

Submodularity:



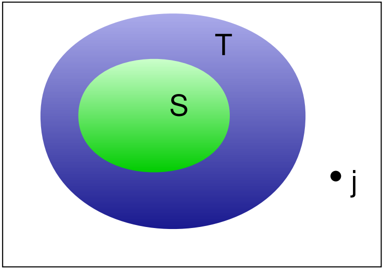
$f : \{0,1\}^n \to \mathbb{R}$ is submodular,

if $\forall i$, the discrete derivative $\partial_i f(x) = f(x + e_i) - f(x)$ is non-increasing in x.

Equivalent definitions

(1) Define the *marginal value of element j*,

$$f_S(j) = f(S \cup \{j\}) - f(S).$$



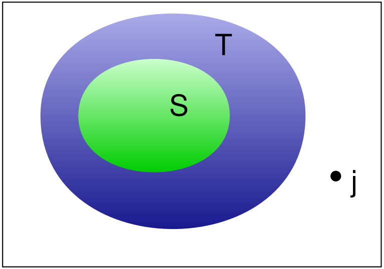
f is submodular, if $\forall S \subset T, j \notin T$:

$$f_S(j) \geq f_T(j).$$

Equivalent definitions

(1) Define the *marginal value of element j*,

$$f_S(j) = f(S \cup \{j\}) - f(S).$$



f is submodular, if $\forall S \subset T, j \notin T$:

$$f_S(j) \geq f_T(j).$$

(2) A function $f : 2^{[n]} \to \mathbb{R}$ is submodular if for any S, T,

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T).$$

Where do submodular functions appear?

1. Foundations of combinatorial optimization:

[Edmonds, Lovász, Schrijver... 70's-90's]

rank functions of matroids, polymatroids, matroid intersection, submodular flows, submodular minimization $\rightarrow$ submodular functions often appear in the background of P-time solvable problems.

Where do submodular functions appear?

1. Foundations of combinatorial optimization:

[Edmonds, Lovász, Schrijver... 70's-90's]

rank functions of matroids, polymatroids, matroid intersection, submodular flows, submodular minimization $\rightarrow$ submodular functions often appear in the background of P-time solvable problems.

2. Algorithmic game theory:

[Lehmann, Lehmann, Nisan, Dobzinski, Papadimitriou, Kempe, Kleinberg, Tardos,... 2000-now]

submodular functions model *valuation functions* of agents with diminishing returns $\rightarrow$ algorithms and incentive-compatible mechanisms for problems like combinatorial auctions, cost sharing, and marketing on social networks.

Where do submodular functions appear?

1. Foundations of combinatorial optimization:

[Edmonds, Lovász, Schrijver... 70's-90's]

rank functions of matroids, polymatroids, matroid intersection, submodular flows, submodular minimization $\rightarrow$ submodular functions often appear in the background of P-time solvable problems.

2. Algorithmic game theory:

[Lehmann, Lehmann, Nisan, Dobzinski, Papadimitriou, Kempe, Kleinberg, Tardos,... 2000-now]

submodular functions model *valuation functions* of agents with diminishing returns $\rightarrow$ algorithms and incentive-compatible mechanisms for problems like combinatorial auctions, cost sharing, and marketing on social networks.

3. Machine learning:

[Guestrin, Krause, Gupta, Golovin, Bilmes,... 2005-now]

submodular functions often appear as objective functions of machine learning tasks such as sensor placement, document summarization or active learning $\rightarrow$ simple algorithms such as Greedy or Local Search work well.

Outline

1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Continuous relaxations for submodular optimization problems.
4. Hardness from symmetric instances.
5. Where do we go next...

Submodular = concave or convex?

- **Argument for concavity:** Definition looks more like concavity - *non-increasing* discrete derivatives.
- **Argument for convexity:** Submodularity seems to be more useful for *minimization* than maximization.

Submodular = concave or convex?

- **Argument for concavity:** Definition looks more like concavity - *non-increasing* discrete derivatives.
- **Argument for convexity:** Submodularity seems to be more useful for *minimization* than maximization.

Theorem (Grötschel-Lovász-Schrijver, 1981; Iwata-Fleischer-Fujishige / Schrijver, 2000)

There is an algorithm that computes the minimum of any submodular function $f : \{0,1\}^n \to \mathbb{R}$ *in* $poly(n)$ *time (using value queries,* $f(S) = ?$*).*

Convex aspects of submodular functions

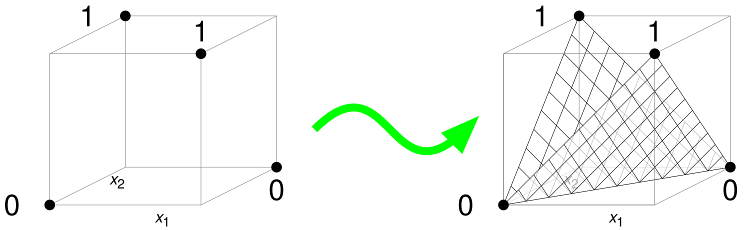
Why is it possible to minimize submodular functions?

- The combinatorial algorithms are sophisticated...
- But there is a simple explanation: the *Lovász extension*.

Convex aspects of submodular functions

Why is it possible to minimize submodular functions?

- The combinatorial algorithms are sophisticated...
- But there is a simple explanation: the *Lovász extension*.



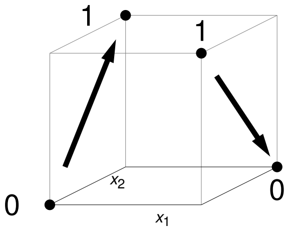
- Submodular function $f \longrightarrow$ convex function f^L,

$$f^L(x) = \mathbb{E}_{\lambda \in [0,1]}[f(\{i : x_i > \lambda\})].$$

- f^L can be minimized efficiently.
- A minimizer of $f^L(x)$ can be converted into a minimizer of $f(S)$.

Concave aspects?

Recall definition: *non-increasing discrete derivatives*.



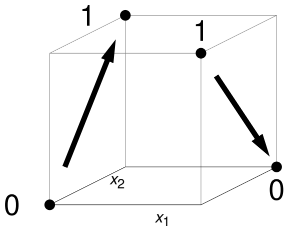
$f : \{0,1\}^n \to \mathbb{R}$ is submodular,

if $\forall i$, the discrete derivative
$\partial_i f(x) = f(x + e_i) - f(x)$
is non-increasing in x.

- Looks like concavity.

Concave aspects?

Recall definition: *non-increasing discrete derivatives*.



$f : \{0,1\}^n \to \mathbb{R}$ is submodular,

if $\forall i$, the discrete derivative
$\partial_i f(x) = f(x + e_i) - f(x)$
is non-increasing in x.

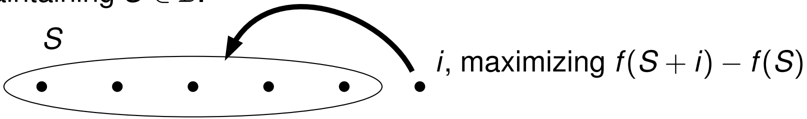
- Looks like concavity.
- But problems involving maximization of submodular functions are typically NP-hard! (Max Cut, Max Coverage,)

So what's going on?

The Greedy Algorithm

Problems in the form: $\max\{f(S) : S \in \mathcal{I}\}$
where f is monotone (non-decreasing) submodular.
$(S \subset T \Rightarrow f(S) \leq f(T))$

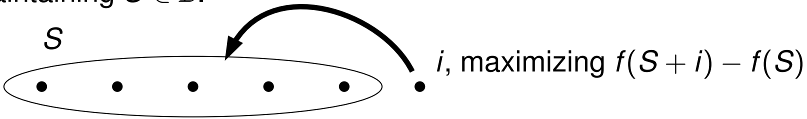
The Greedy Algorithm: [Nemhauser,Wolsey,Fisher '78]
Pick elements one-by-one, maximizing the gain in $f(S)$, while maintaining $S \in \mathcal{I}$.



The Greedy Algorithm

Problems in the form: $\max\{f(S) : S \in \mathcal{I}\}$
where f is monotone (non-decreasing) submodular.
$(S \subset T \Rightarrow f(S) \leq f(T))$

The Greedy Algorithm: [Nemhauser,Wolsey,Fisher '78]
Pick elements one-by-one, maximizing the gain in $f(S)$, while maintaining $S \in \mathcal{I}$.



Theorem (Nemhauser,Wolsey,Fisher '78)

If f is monotone submodular, Greedy finds a solution of value at least $(1 - 1/e)\times$ optimum for the problem $\max\{f(S) : |S| \leq k\}$.

Optimality: [NW'78] *No algorithm using a polynomial number of queries to f can achieve a factor better than* $1 - 1/e$.

Applications of the Greedy Algorithm

Greedy Algorithm provides: (using [NWF '78])

1. $(1 - 1/e)$-approximation for various problems in optimization and machine learning in the form $\max\{f(S) : |S| \leq k\}$;

   marketing on social networks [Kempe-Kleinberg-Tardos '03], optimal sensor placement [Krause-Guestrin-et al. '06-'10]

2. $1/2$-approximation for problems in the form $\max\{f(S) : S \in \mathcal{I}\}$ where $\mathcal{I}$ is a matroid;

   e.g. welfare maximization in combinatorial auctions with submodular bidders [Lehmann-Lehmann-Nisan '01]

Applications of the Greedy Algorithm

Greedy Algorithm provides: (using [NWF '78])

1. $(1 - 1/e)$-approximation for various problems in optimization and machine learning in the form $\max\{f(S) : |S| \leq k\}$;

   marketing on social networks [Kempe-Kleinberg-Tardos '03], optimal sensor placement [Krause-Guestrin-et al. '06-'10]

2. $1/2$-approximation for problems in the form $\max\{f(S) : S \in \mathcal{I}\}$ where $\mathcal{I}$ is a matroid;

   e.g. welfare maximization in combinatorial auctions with submodular bidders [Lehmann-Lehmann-Nisan '01]

Questions that don't seem to be answered by the greedy algorithm:

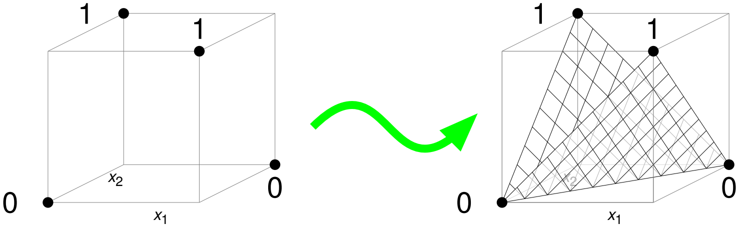
- Optimal approximation for $\max\{f(S) : S \in \mathcal{I}\}$ where f is monotone submodular and $\mathcal{I}$ forms a matroid
- Optimization of *non-monotone* submodular functions
- More general constraints, or combinations of simple ones

Outline

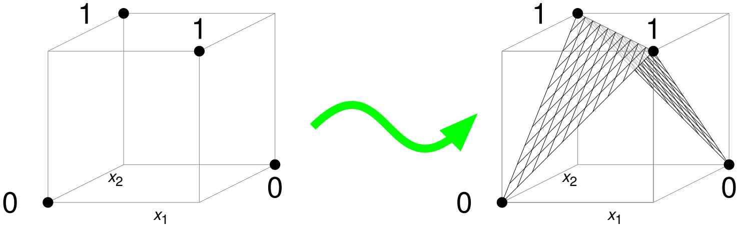
1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Continuous relaxations for submodular optimization problems.
4. Hardness from symmetric instances.
5. Where do we go next...

Continuous relaxation for submodular maximization?

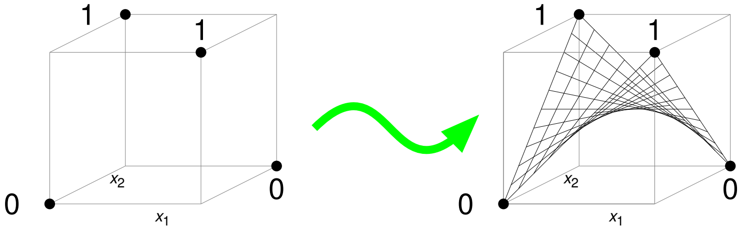
1. The *Lovász extension* is convex — not suitable for maximization.



2. There is also a "concave closure". However, NP-hard to evaluate!



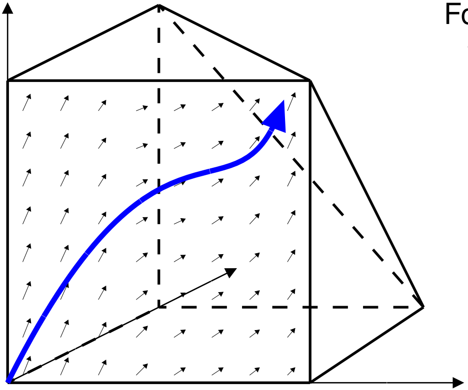
Multilinear extension of f:



- $F(x) = \mathbb{E}[f(\hat{x})]$, where $\hat{x}$ is obtained by rounding each x_i randomly to $0/1$ with probabilities x_i.
- $F(x)$ is neither convex nor concave; it is multilinear and $\frac{\partial^2 F}{\partial {x_i}^2} = 0$.
- $F(x + \lambda \vec{d})$ is a *concave* function of λ, if $\vec{d} \geq 0$.

The Continuous Greedy Algorithm [V. '08]

Problem: $\max\{F(x) : x \in P\}$,
F multilinear extension of a *monotone submodular function*.



For each $x \in P$, define $v(x)$ by
$v(x) = argmax_{v \in P}(v \cdot \nabla F|_x)$.

Define a curve $x(t)$:

$$x(0) = 0$$

$$\frac{dx}{dt} = v(x)$$

Run this process
for $t \in [0, 1]$ and return $x(1)$.

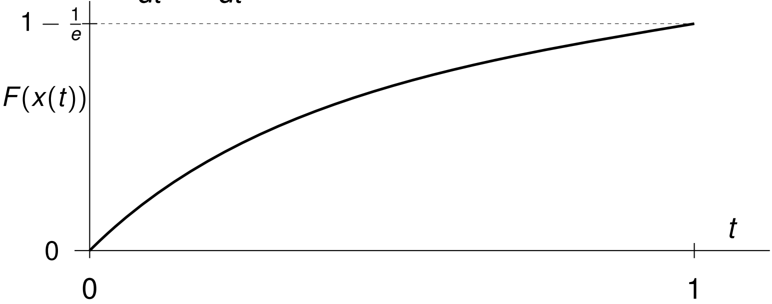
Claim: $x(1) \in P$ and $F(x(1)) \geq (1 - 1/e)OPT$.

Analysis of Continuous Greedy

Evolution of the fractional solution:

- Differential equation: $x(0) = 0$, $\frac{dx}{dt} = v(x)$.
- Chain rule:

$$\frac{dF}{dt} = \frac{dx}{dt} \cdot \nabla F(x(t)) = v(x) \cdot \nabla F(x(t)) \geq OPT - F(x(t)).$$



Solve the differential equation:

$$F(x(t)) \geq (1 - e^{-t}) \cdot OPT.$$

Applications of the multilinear relaxation

Algorithms:

1. $(1 - 1/e)$-approximation for the problem $\max\{f(S) : S \in \mathcal{I}\}$ where f is monotone submodular and $\mathcal{I}$ is a matroid [Calinescu-Chekuri-Pál-V. '08]
2. $(1 - 1/e - \epsilon)$-approximation for maximizing subject to $O(1)$ linear constraints: $\max\{f(S) : \forall i; \sum_{j \in S} c_{ij} \leq 1\}$ [Kulik-Shachnai-Tamir '09]
3. $1/e$-approximation for the same problems with non-monotone submodular valuations [Feldman-Naor-Schwartz '11]
4. Approximations for more general constraints [Chekuri-Zenklusen-V. '11]

Applications of the multilinear relaxation

Algorithms:

1. $(1 - 1/e)$-approximation for the problem $\max\{f(S) : S \in \mathcal{I}\}$ where f is monotone submodular and $\mathcal{I}$ is a matroid [Calinescu-Chekuri-Pál-V. '08]
2. $(1 - 1/e - \epsilon)$-approximation for maximizing subject to $O(1)$ linear constraints: $\max\{f(S) : \forall i; \sum_{j \in S} c_{ij} \leq 1\}$ [Kulik-Shachnai-Tamir '09]
3. $1/e$-approximation for the same problems with non-monotone submodular valuations [Feldman-Naor-Schwartz '11]
4. Approximations for more general constraints [Chekuri-Zenklusen-V. '11]

Open question: What is the best approximation for the problem $max\{F(x) : x \in P\}$ when F is non-monotone submodular?

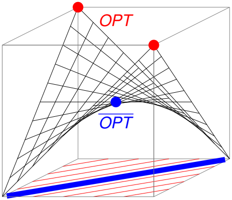
(We know a $1/e \simeq 0.36$-approximation, and better than 0.48 is impossible.)

Outline

1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Continuous relaxations for submodular optimization problems.
4. Hardness from symmetric instances.
5. Where do we go next...

Hardness from multilinear relaxation

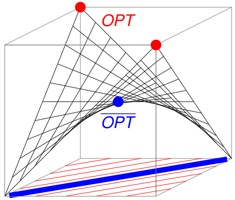
Symmetry gap: ratio $\gamma = \overline{OPT}/OPT$ between the best *symmetric* and the best *asymmetric* solution for the multilinear relaxation



$$OPT = \max\{F(x) : x \in P\}$$
$$\overline{OPT} = \max\{F(x) : x \in P_{sym}\}$$

Hardness from multilinear relaxation

Symmetry gap: ratio $\gamma = \overline{OPT}/OPT$ between the best *symmetric* and the best *asymmetric* solution for the multilinear relaxation



$$OPT = \max\{F(x) : x \in P\}$$
$$\overline{OPT} = \max\{F(x) : x \in P_{sym}\}$$

Idea: Based on such an example, we can generate similar instances such that an efficient algorithm will find only "symmetric solutions", hence it cannot get close to the optimum.

Hardness from symmetry gap

Theorem (informal)

For an instance of submodular optimization with symmetry gap γ and any $\epsilon > 0$, there is a family of "similar" instances $\mathcal{F}$ such that no efficient algorithm can achieve a $(\gamma + \epsilon)$-approximation on $\mathcal{F}$.

(using value queries [V. '09], or on explicit instances assuming that NP $\neq$ RP [Dobzinski, V. '12])

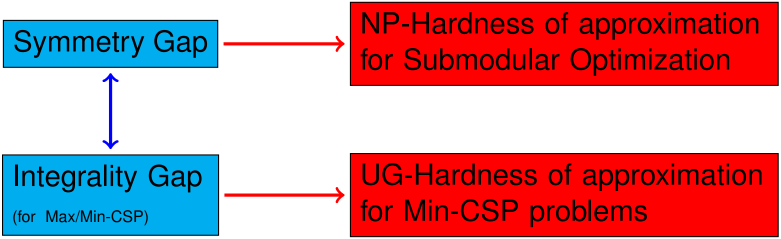
Consequences: unification of known hardness results, some new results.

- $1 - 1/e$ is optimal for $\max\{f(S) : |S| \leq k\}$, f monotone [Nemhauser,Wolsey '78]
- $1/2$ is optimal for $\max\{f(S) : S \subseteq N\}$, f non-monotone submodular [Feige,Mirrokni,V. '06][Buchbinder,Feldman,Naor,Schwartz '12]
- 0.49 cannot be achieved for $\max\{f(S) : |S| \leq k\}$, f non-monotone submodular [Oveis Gharan, V. '11]
- no constant factor can be achieved for $\max\{f(S) : S \in \mathcal{B}\}$, f non-monotone submodular and $\mathcal{B}$ bases in a matroid [V. '09]

Connections with integrality gap and UG-hardness

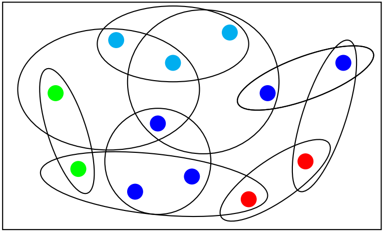
- Symmetry gap implies hardness for a submodular optimization problem: in the oracle model [V. '09] and also NP-hardness for explicit instances [Dobzinski,V. '12].
- It is equal to the integrality gap of a related LP formulation.
- For *Min-CSP problems with the Not-Equal predicate*, this also implies hardness assuming the *Unique-Games Conjecture*.

[Manokaran,Naor,Raghavendra,Schwartz '08] [Ene,V.,Wu '13]



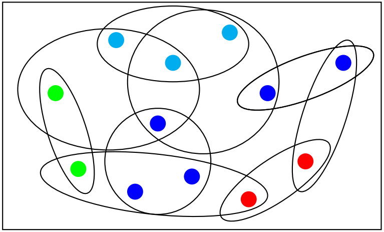
Example: a Hypergraph Labeling Problem

Hypergraph Labeling: (problem in Min-CSP form)
Given a hypergraph $H = (V, E)$ with color lists $L(v) \subseteq [k]\ \forall v \in V$, find a coloring $\ell(v) \in L(v)$ that minimizes the number of hyperedges with more than 1 color.



Example: a Hypergraph Labeling Problem

Hypergraph Labeling: (problem in Min-CSP form)
Given a hypergraph $H = (V, E)$ with color lists $L(v) \subseteq [k]\ \forall v \in V$, find a coloring $\ell(v) \in L(v)$ that minimizes the number of hyperedges with more than 1 color.

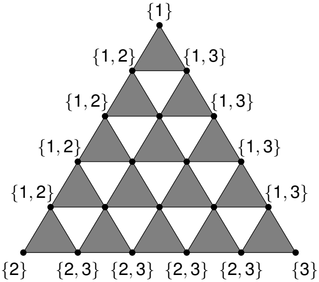


Submodular generalization: Minimize $\sum_{i=1}^{k} f(\ell^{-1}(i))$

- Lovász extension gives a natural convex relaxation
- $\longrightarrow$ k-approximation for Hypergraph Labeling [Chekuri-Ene '11]
- is this best possible? enough to find a symmetry/integrality gap

Integrality Gap example for Hypergraph Labeling

Sperner-style setup: hyperedges are k-vertex simplices, scaled copies of a large simplex; vertices on the boundary are allowed to use only the colors of the respective face.

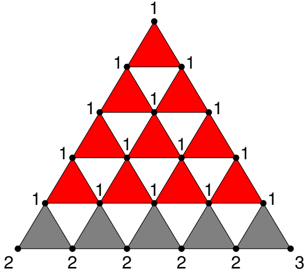


Question: [Ene-V. '14]

What is the minimum possible number of *non-monochromatic hyperedges*?

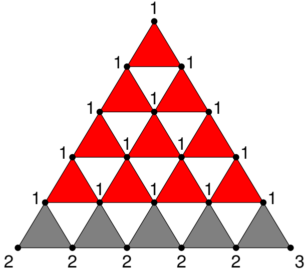
Optimal Coloring => Integrality gap

Plausible conjecture: the following "simple coloring" has the minimum number of non-monochromatic hyperedges:



Optimal Coloring => Integrality gap

Plausible conjecture: the following "simple coloring" has the minimum number of non-monochromatic hyperedges:

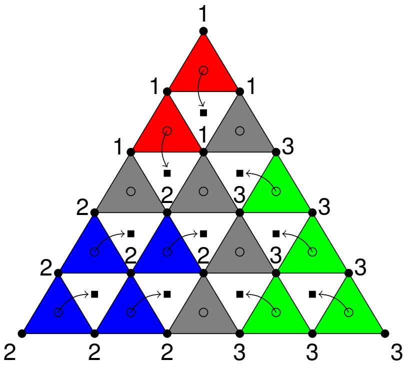


This gives an integrality gap:

- Total # hyperedges $\simeq \frac{1}{(k-1)!} n^k$
- LP cost $\simeq \frac{1}{n} \cdot \frac{1}{(k-1)!} n^k = \frac{1}{(k-1)!} n^{k-1}$
- Optimum cost $\simeq \frac{1}{(k-2)!} n^{k-1} = (k-1) \times$ *LP cost*
- this would imply that there is no approximation better than $(k-1)$, assuming the Unique Games Conjecture (using [Ene-V.-Wu '13])

Proof of the optimality of the "simple coloring"

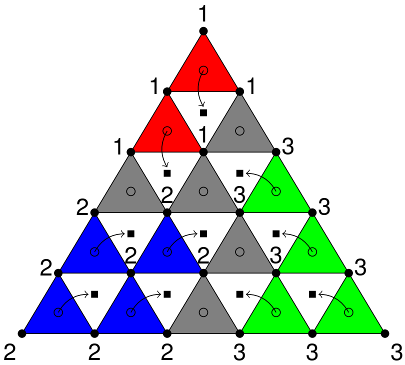
[Mirzakhani '14]



- Set of vertices $= V_{n,k}$, hyperedges can be identified with $V_{n-1,k}$
- Map *monochromatic hyperedges* to $V_{n-2,k}$ as in the picture.

Proof of the optimality of the "simple coloring"

[Mirzakhani '14]



- Set of vertices $= V_{n,k}$, hyperedges can be identified with $V_{n-1,k}$
- Map *monochromatic hyperedges* to $V_{n-2,k}$ as in the picture.
- Since two monochromatic hyperedges cannot be neighbors, the mapping is injective.
- $\#\text{non-mono}\chi \geq |V_{n-1,k}| - |V_{n-2,k}| = \binom{n+k-2}{k-1} - \binom{n+k-3}{k-1} = \binom{n+k-3}{k-2}$.

Future directions

We have:

Discrete optimization of linear functions $\leftrightarrow$ Linear Programming
Minimization of submodular functions $\leftrightarrow$ Lovász Relaxation
Maximization of submodular functions $\leftrightarrow$ Multilinear Relaxation

Future directions

We have:

Discrete optimization of linear functions $\leftrightarrow$ Linear Programming
Minimization of submodular functions $\leftrightarrow$ Lovász Relaxation
Maximization of submodular functions $\leftrightarrow$ Multilinear Relaxation

Fundamental questions:

1. Can we solve every maximization problem with a monotone submodular objective that we can solve with a linear objective (up to constant factors)?
2. The best approximations are achieved by randomized algorithms - is randomization necessary?

Future directions

We have:

Discrete optimization of linear functions $\leftrightarrow$ Linear Programming
Minimization of submodular functions $\leftrightarrow$ Lovász Relaxation
Maximization of submodular functions $\leftrightarrow$ Multilinear Relaxation

Fundamental questions:

1. Can we solve every maximization problem with a monotone submodular objective that we can solve with a linear objective (up to constant factors)?
2. The best approximations are achieved by randomized algorithms - is randomization necessary?

Practical question: some progress in [Ashwinkumar-V. '14]

1. Can these algorithms be used in practice — better running times?