Submodular Functions and Their Applications

Jan Vondrák[1]

[1]IBM Almaden Research Center
San Jose, CA

ACM-SIAM SODA, New Orleans, LA
January 7, 2013

Prelude: what is this about?

There are two kinds of mathematicians:

- **The Problem Solver:** "How can I solve this problem?"
- **The Theory Builder:** "What is the structure that would allow me to solve more and more problems?"

Prelude: what is this about?

There are two kinds of mathematicians:

- **The Problem Solver:** "How can I solve this problem?"
- **The Theory Builder:** "What is the structure that would allow me to solve more and more problems?"

Problem Solvers $\longrightarrow$ *combinatorics*...

Theory Builders $\longrightarrow$ *algebra*...

Prelude: what is this about?

There are two kinds of mathematicians:

- **The Problem Solver:** "How can I solve this problem?"
- **The Theory Builder:** "What is the structure that would allow me to solve more and more problems?"

Problem Solvers $\longrightarrow$ *combinatorics*...

Theory Builders $\longrightarrow$ *algebra*...

Dominic Welsh [1976]:
"... mathematical generalization often lays bare the important bits of information about the problem at hand."

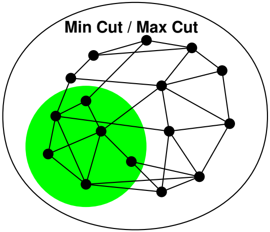
Outline

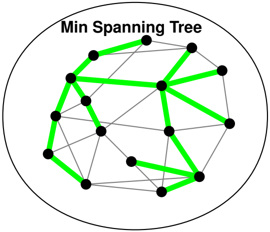
1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Applications of submodular maximization.
4. Recent advances: the multilinear relaxation.

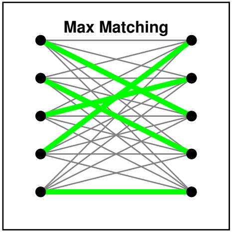
Discrete optimization

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:

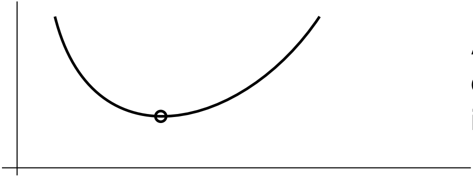
Traveling Salesman, Max Clique, Max Cut, Set Cover, Knapsack,...

Continuous optimization

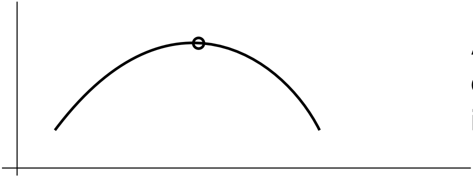
What makes *continuous optimization* tractable?

Continuous optimization

What makes *continuous optimization* tractable?



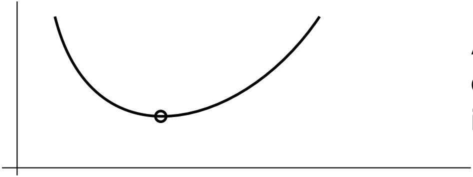
A function $f : \mathbb{R}^n \to \mathbb{R}$
can be minimized efficiently,
if it is convex.



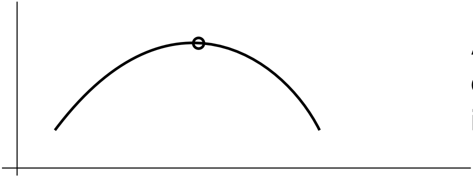
A function $f : \mathbb{R}^n \to \mathbb{R}$
can be maximized efficiently,
if it is concave.

Continuous optimization

What makes *continuous optimization* tractable?



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



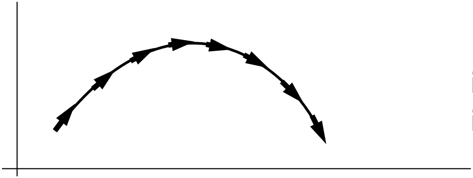
A function $f : \mathbb{R}^n \to \mathbb{R}$
can be maximized efficiently,
if it is concave.

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

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

From concavity to submodularity

Concavity:

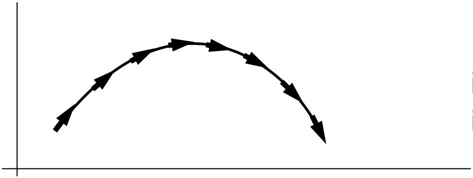


$f : \mathbb{R} \to \mathbb{R}$ is concave,

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

From concavity to submodularity

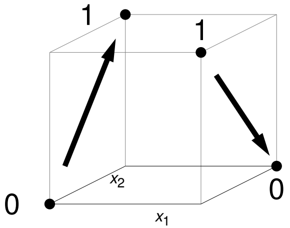
Concavity:



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

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

Submodularity:



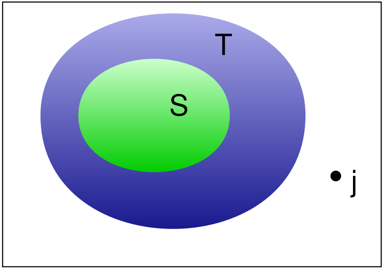
$f : \{0,1\}^n \to \mathbb{R}$ is submodular,

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

Equivalent definitions

(1) For $f : 2^{[n]} \to \mathbb{R}$, define the *marginal value of element j*,

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



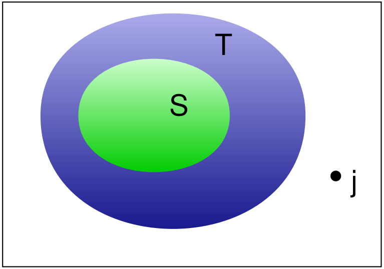
f is submodular, if $\forall S \subset T, j \notin T$:

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

Equivalent definitions

(1) For $f : 2^{[n]} \to \mathbb{R}$, define the *marginal value of element j*,

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



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

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

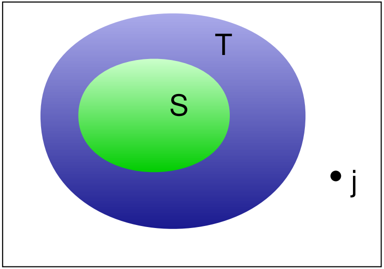
(2) $f : 2^{[n]} \to \mathbb{R}$ is submodular if for any S, T,

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

Equivalent definitions

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

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

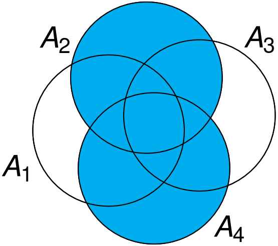
Value oracle model

Access to f: through an oracle answering queries "$f(S) = ?$"

Examples of submodular functions

Coverage function:
Given $A_1, \ldots, A_n \subset U$,

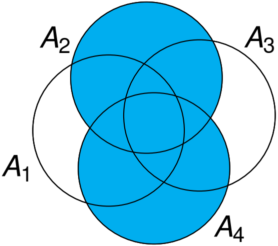
$$f(S) = \left| \bigcup_{j \in S} A_j \right|.$$

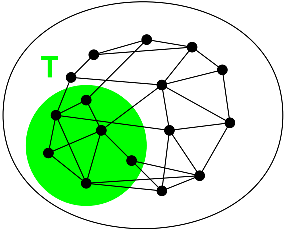


Examples of submodular functions

Coverage function:
Given $A_1, \ldots, A_n \subset U$,

$$f(S) = \left| \bigcup_{j \in S} A_j \right|.$$





Cut function:

$$\delta(T) = |e(T, \overline{T})|$$

Outline

1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Applications of submodular maximization.
4. Recent advances: the multilinear relaxation.

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 (in the value oracle model).*

Convex aspects of submodular functions

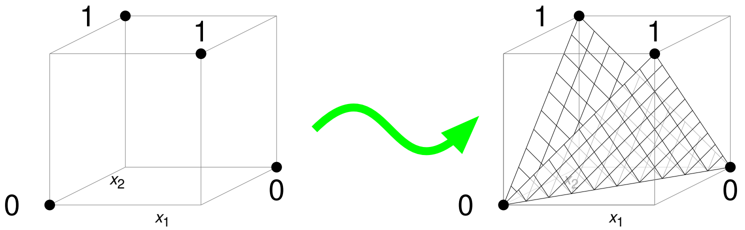
Why is it possible to minimize submodular functions?

- 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*.



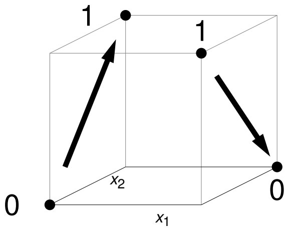
- Submodular function $f \longrightarrow$ convex function f^L,

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

- 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*.



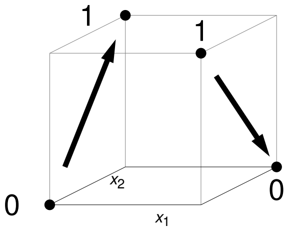
$f : \{0,1\}^n \to \mathbb{R}$ is submodular,

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*.



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

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

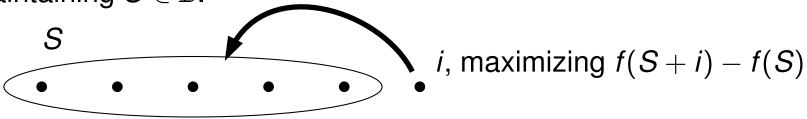
- Looks like concavity.
- But problems involving maximization of submodular functions are typically NP-hard! (Max Cut, Max Coverage)

So what's going on?

The Greedy Algorithm

Problems in the form: $\max\{f(S) : S \in \mathcal{I}\}$
where $f : 2^N \to \mathbb{R}_+$ is monotone (non-decreasing) submodular.

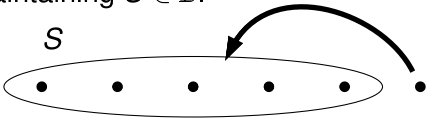
The Greedy Algorithm: [Nemhauser-Wolsey-Fisher '78]
Pick elements one-by-one, maximizing the gain in $f(S)$, while maintaining $S \in \mathcal{I}$.



The Greedy Algorithm

Problems in the form: $\max\{f(S) : S \in \mathcal{I}\}$
where $f : 2^N \to \mathbb{R}_+$ is monotone (non-decreasing) submodular.

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



i, maximizing $f(S + i) - f(S)$

Theorem (Nemhauser,Wolsey,Fisher '78)

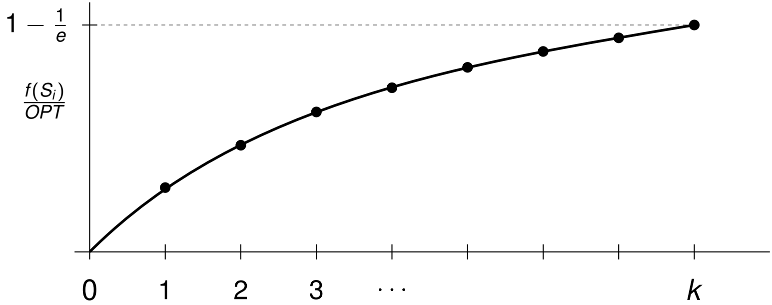
If f is monotone submodular, Greedy finds a solution of value at least $(1 - 1/e)\times$ optimum for the problem $\max\{f(S) : |S| \leq k\}$.

(and this is best possible for this problem) [Nemhauser-Wolsey '78] [Feige '98]

Greedy Algorithm: *S_i = solution after i steps; pick next element a to maximize $f(S_i + a) - f(S_i)$.*

Let the optimal solution be S^*. By submodularity:

$$\exists a \in S^* \setminus S_i;\ f(S_i + a) - f(S_i) \geq \frac{1}{k}(OPT - f(S_i)).$$



$$OPT - f(S_{i+1}) \leq (1 - \tfrac{1}{k})(OPT - f(S_i))$$
$$\Rightarrow OPT - f(S_k)) \leq (1 - \tfrac{1}{k})^k OPT \leq \tfrac{1}{e} OPT.$$

Outline

1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Applications of submodular maximization.
4. Recent advances: the multilinear relaxation.

Applications of Submodular Maximization

Spreading of Influence in Social Networks [Kempe-Kleinberg-Tardos '03]
$(1 - 1/e)$-approximation for choosing an initial set of active nodes, to maximize the expected number of nodes activated at the end

Sensor Placement in Machine Learning [Guestrin-Krause et al.]
$(1 - 1/e)$-approximation for placing a set of sensors in order to extract maximum amount of information in various settings
(using the submodularity of the *entropy function*)

Battle of Water Sensor Networks (Water Distribution Systems Analysis Symposium, Cincinnati, 2006)
1st prize using a greedy-based algorithm for detecting contamination outbreaks in a water network [Leskovec-Krause-Guestrin-Faloutsos-VanBriesen-Glance]

Document Summarization [Bilmes-Lin '11]
Algorithms for extracting a small collection of phrases/sentences to represent a large document (using submodularity of a coverage+diversity function)

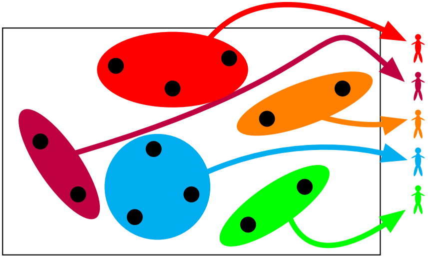
Algorithmic Game Theory

Submodular functions $\simeq$ valuations with diminishing returns

Algorithmic Game Theory

Submodular functions $\simeq$ valuations with diminishing returns

Combinatorial auctions [Lehmann-Lehmann-Nisan '01]
$|M| = m$ items are to be sold to n agents
with (monotone) valuations $v_i : 2^M \to \mathbb{R}_+$.



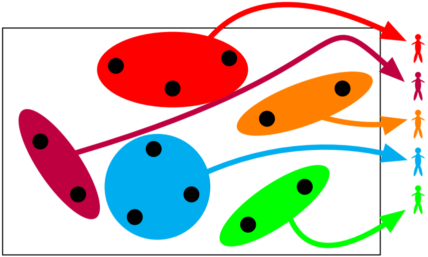
Allocate S_i to agent i

to maximize $\sum_{i=1}^{n} v_i(S_i)$.

Algorithmic Game Theory

Submodular functions $\simeq$ valuations with diminishing returns

Combinatorial auctions [Lehmann-Lehmann-Nisan '01]
$|M| = m$ items are to be sold to n agents
with (monotone) valuations $v_i : 2^M \to \mathbb{R}_+$.



Allocate S_i to agent i

to maximize $\sum_{i=1}^{n} v_i(S_i)$.

Submodular Welfare Maximization [Nemhauser-Wolsey-Fisher '78]
for *submodular valuations*, Greedy gives a $1/2$-approximation.

Beyond the Greedy Algorithm

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

- *Optimal* approximations (Submodular Welfare Problem?)
- *Non-monotone* submodular functions?
- Extensions to *more general constraints*?

Beyond the Greedy Algorithm

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

- *Optimal* approximations (Submodular Welfare Problem?)
- *Non-monotone* submodular functions?
- Extensions to *more general constraints*?

Alternative techniques:

Greedy algorithm with partial enumeration [Sviridenko '04]

Iterated greedy algorithms [Gupta-Roth-Schoenebeck-Talwar '09]

Local search algorithms [Feige-Mirrokni-V.'07] [Lee-Mirrokni-Nagarajan-Sviridenko'09] [Lee-Sviridenko-V.'09] [Feldman-Naor-Schwartz-Ward '11] [Filmus-Ward '12]

Beyond the Greedy Algorithm

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

- *Optimal* approximations (Submodular Welfare Problem?)
- *Non-monotone* submodular functions?
- Extensions to *more general constraints*?

Alternative techniques:
Greedy algorithm with partial enumeration [Sviridenko '04]
Iterated greedy algorithms [Gupta-Roth-Schoenebeck-Talwar '09]
Local search algorithms [Feige-Mirrokni-V.'07] [Lee-Mirrokni-Nagarajan-Sviridenko'09] [Lee-Sviridenko-V.'09] [Feldman-Naor-Schwartz-Ward '11] [Filmus-Ward '12]

General question:

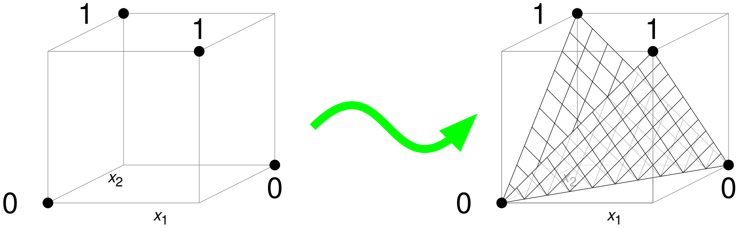
- continuous relaxation for submodular maximization problems?

Outline

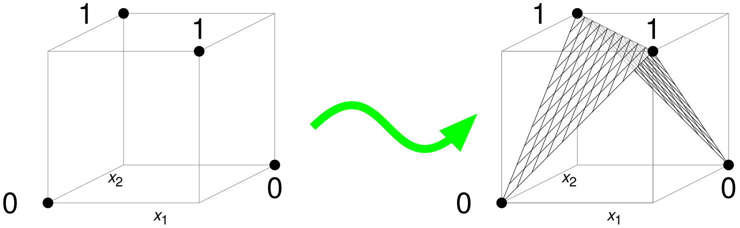
1. What are submodular functions?
2. Is submodularity more like convexity or concavity?
3. Applications of submodular maximization.
4. Recent advances: the multilinear relaxation.

Continuous relaxation for submodular maximization?

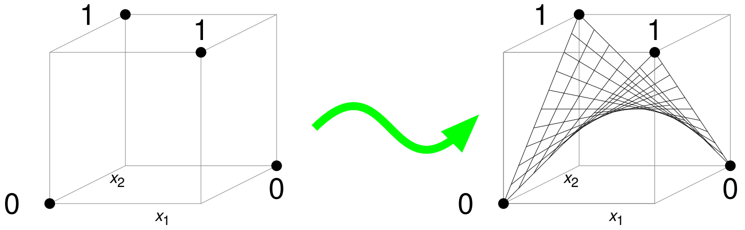
1. The *Lovász extension* is convex — not suitable for maximization.



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.
- $F(x + \lambda \vec{d})$ is a *concave* function of λ, if $\vec{d} \geq 0$.

How to use the multilinear relaxation

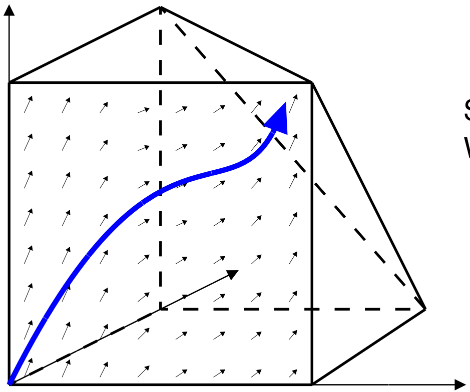
The multilinear relaxation turns out to be useful for maximization:

1. **The continuous problem** $\max\{F(x) : x \in P\}$ can be solved:
   - $(1 - 1/e)$-approximately for any monotone submodular function and solvable polytope [V. '08]
   - $(1/e)$-approximately for any nonnegative submodular function and down-closed solvable polytope [Feldman-Naor-Schwartz '11]
2. **A fractional solution can be rounded:**
   - without loss for a matroid constraint [Calinescu-Chekuri-Pál-V. '07]
   - losing $(1 - \epsilon)$ factor for a constant number of linear constraints [Kulik-Shachnai-Tamir '10]
   - for more general constraints, using *contention resolution schemes* [Chekuri-V.-Zenklusen '11]
   - e.g., $O(k)$-approximation for k matroids & $O(1)$ linear constraints

In particular:
Optimal $(1 - 1/e)$-approximation for the Submodular Welfare Problem.

The Continuous Greedy Algorithm [V. '08]

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



Start with $x = \mathbf{0}, t = 0$;
While $t < 1$
  Find $v \in P$ maximizing $v \cdot \nabla F|_x$;
  Update $x = x + \delta v, \;\; t = t + \delta$;

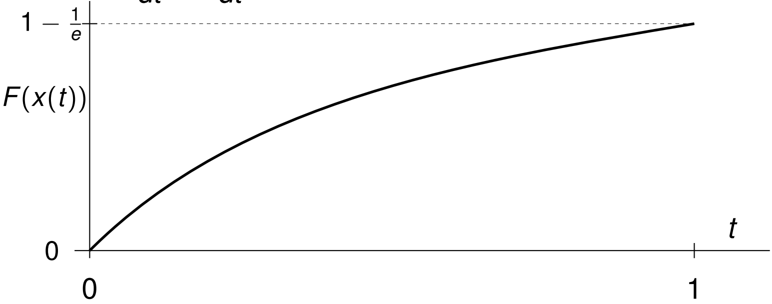
Claim: $x(1) \in P$ and $F(x(1)) \geq (1 - 1/e)OPT$.

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:

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

Multilinear relaxation for non-monotone functions

Non-monotone submodular functions: [Feldman-Naor-Schwartz '11]

- Continuous greedy still solves $\max\{F(x) : x \in P\}$ within a constant factor for any down-closed P.
- A careful modification achieves a factor of $1/e$; e.g., for Submodular Welfare Problem with *non-monotone* valuations.

Multilinear relaxation for non-monotone functions

Non-monotone submodular functions: [Feldman-Naor-Schwartz '11]

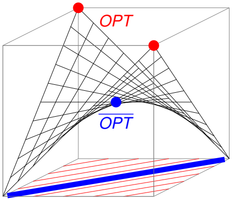
- Continuous greedy still solves $\max\{F(x) : x \in P\}$ within a constant factor for any down-closed P.
- A careful modification achieves a factor of $1/e$; e.g., for Submodular Welfare Problem with *non-monotone* valuations.

Unconstrained submodular maximization: [Buchbinder-Feldman-Naor-Schwartz '12]

- $1/2$-approximation for the problem $\max_{S \subseteq N} f(S)$
  — *double-greedy randomized algorithm*
- No $(1/2 + \epsilon)$-approx in the value oracle model [Feige-Mirrokni-V. '07]
  — hardness can be derived from the multilinear relaxation

Hardness from multilinear relaxation

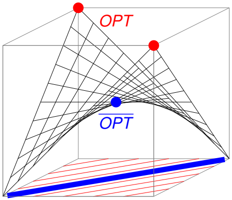
Symmetry gap: ratio $\gamma = \frac{\overline{OPT}}{OPT}$ between the best *symmetric* and the best *asymmetric* solution for the multilinear relaxation of an instance $\max\{F(x) : x \in P\}$.



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

Hardness from multilinear relaxation

Symmetry gap: ratio $\gamma = \frac{\overline{OPT}}{OPT}$ between the best *symmetric* and the best *asymmetric* solution for the multilinear relaxation of an instance $\max\{F(x) : x \in P\}$.



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

Hardness result:
no $(\gamma + \epsilon)$-approximation for instances "of the same type"
(oracle hardness [V. '09], computational hardness [Dobzinski-V. '12])

E.g.: symmetry gap for unconstrained maximization is $1/2$
$\Rightarrow$ no approximation better than $1/2$ for this problem.

Submodular maximization overview

MONOTONE MAXIMIZATION

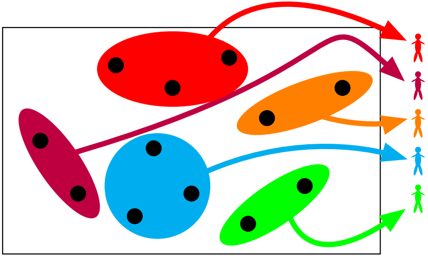
Constraint	Approximation	Hardness	technique
$\|S\| \leq k$	$1 - 1/e$	$1 - 1/e$	greedy
matroid	$1 - 1/e$	$1 - 1/e$	multilinear ext.
$O(1)$ knapsacks	$1 - 1/e$	$1 - 1/e$	multilinear ext.
k matroids	$k + \epsilon$	$k/\log k$	local search
k matroids & $O(1)$ knapsacks	$O(k)$	$k/\log k$	multilinear ext.

NON-MONOTONE MAXIMIZATION

Constraint	Approximation	Hardness	technique
Unconstrained	$1/2$	$1/2$	combinatorial
matroid	$1/e$	0.48	multilinear ext.
$O(1)$ knapsacks	$1/e$	0.49	multilinear ext.
k matroids	$k + O(1)$	$k/\log k$	local search
k matroids & $O(1)$ knapsacks	$O(k)$	$k/\log k$	multilinear ext.

Truthful mechanism design

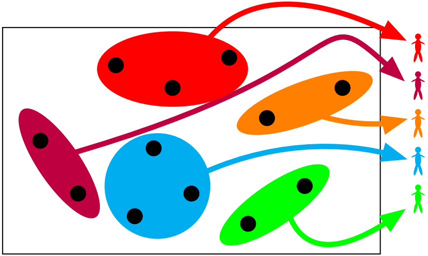
Combinatorial auctions:



Allocate S_i to agent i
to maximize $\sum_{i=1}^{n} v_i(S_i)$
+ incentivize agents
to report their true valuations!

Truthful mechanism design

Combinatorial auctions:



Allocate S_i to agent i
to maximize $\sum_{i=1}^{n} v_i(S_i)$
+ incentivize agents
to report their true valuations!

Building on the multilinear relaxation:

- There is a $(1 - 1/e)$-approximate truthful-in-expectation mechanism for coverage valuations [Dughmi-Roughgarden-Yan '11]
- However, no truthful-in-expectation $m^{o(1)}$-approximation for submodular valuations oracle hardness [Dughmi-V. '11], computational hardness [Dobzinski-V. '12]

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

Main open questions:

1. Can we approximate every maximization problem with a monotone submodular objective (up to constant factors) if we can approximate it with a linear objective?
2. Does the symmetry gap characterize approximability in some sense — "if and only if" results?
3. Beyond submodularity: more general "polymorphisms"?