ETH
Eidgenössische Technische Hochschule Zürich

Submodularity in Data Science

Andreas Krause

Data Science Summer School

Acknowledgments

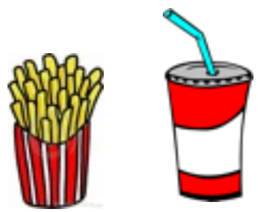
- Based on earlier tutorials with Stefanie Jegelka

Set functions

ground set $\mathcal{V} = \{$  $\}$

$S \subseteq \mathcal{V}$

$F : 2^{\mathcal{V}} \to \mathbb{R}$

$F($  $) =$

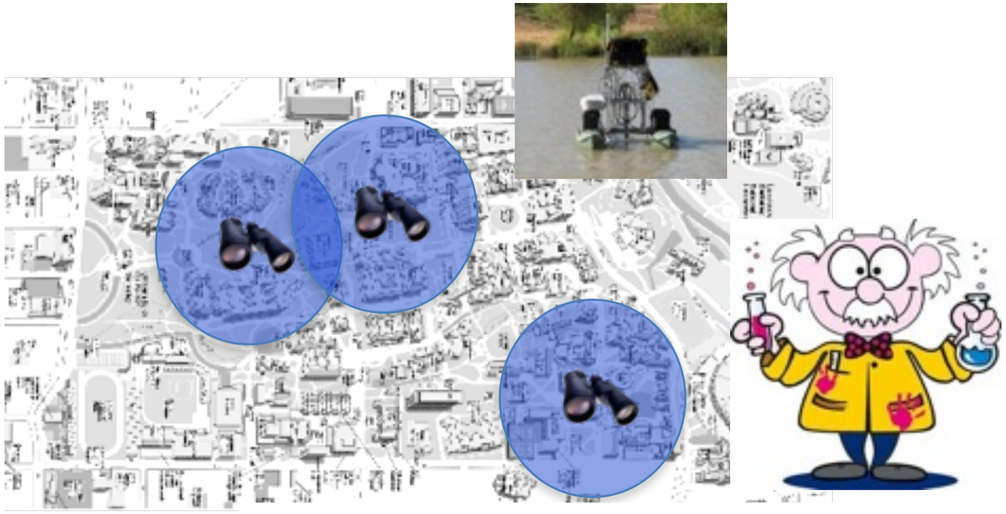
cost/loss of picking items together, or

utility, or

probability, ...

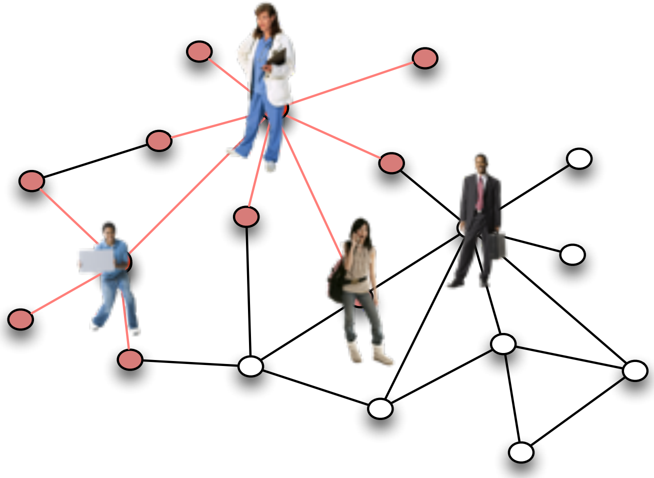
We will assume:

- $F(\emptyset) = 0$
- black box "oracle" to evaluate *F*



$\mathcal{V} =$ variables to observe

$F(S) =$ "information"



$\mathcal{V} =$ seed nodes

$F(S) =$ "spread"

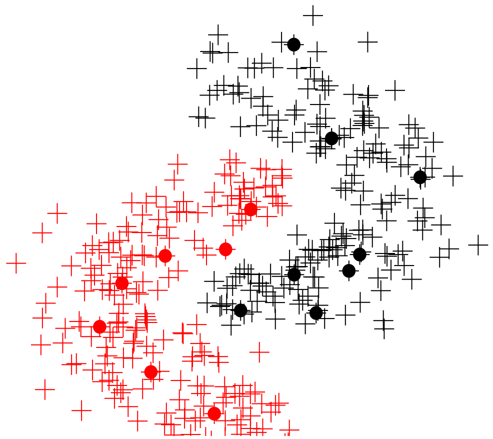
$\mathcal{V} =$ images (sentences, ...)

$F(S) =$ "representation"

Dictionary learning, matrix approximation, object detection,...

maximize coverage, spread, diversity

$$\max_S \; F(S)$$



$\mathcal{V} =$ data points

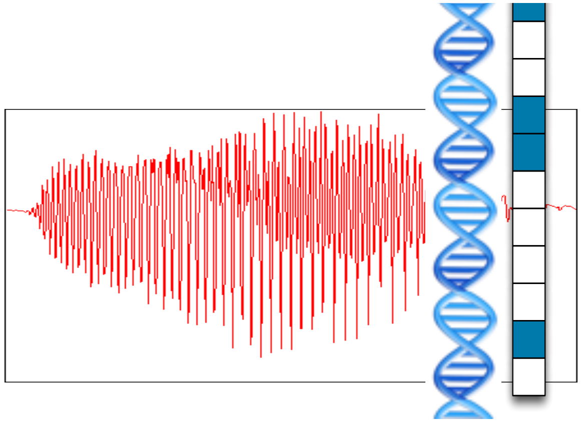
$F(S) =$ "coherence/separation"

$\mathcal{V} =$ pixels

$F(S) =$ "coherence/matching"

$\mathcal{V} =$ coordinates (variables)

$F(S) =$ "coherence"



maximize coherence, smoothness

$\min_S \; F(S)$

Convex functions *(Lovász, 1983)*

- "occur in many models in economy, engineering and other sciences", "often the only nontrivial property that can be stated in general"
- preserved under many operations and transformations: larger effective range of results
- sufficient structure for a "mathematically beautiful and practically useful theory"
- efficient minimization

"It is less apparent, but we claim and hope to prove to a certain extent, that a similar role is played in discrete optimization by *submodular set-functions*" [...]
they share the above four properties.

Outline

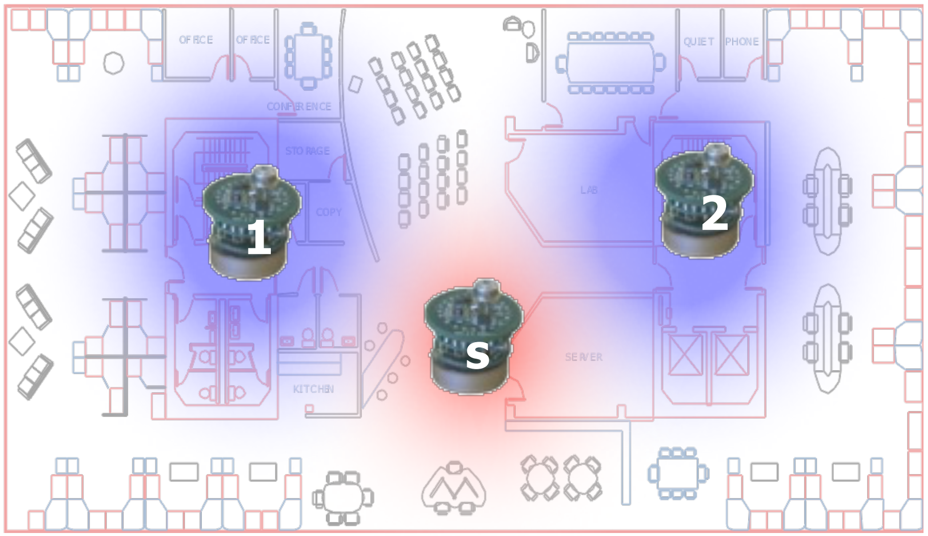
1. What is Submodularity?
   Examples, connections
2. Submodular minimization
3. Submodular maximization

TOMORROW

4. Advanced Topics
   submodularity in deep learning, probabilistic inference, active learning, bandits, …

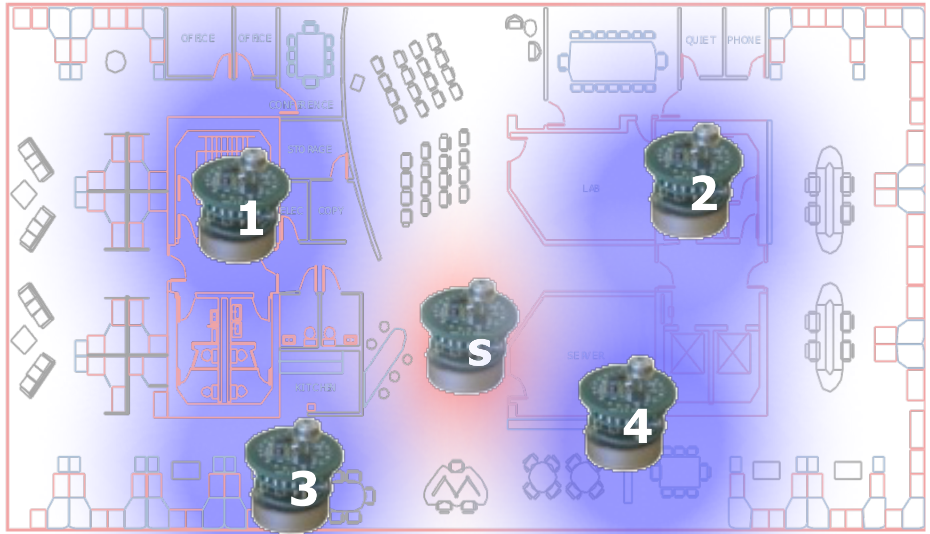
Diminishing gains

placement A = {1,2}



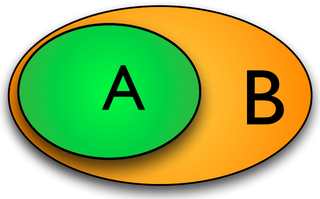
Big gain

placement B = {1,2,3,4}



small gain

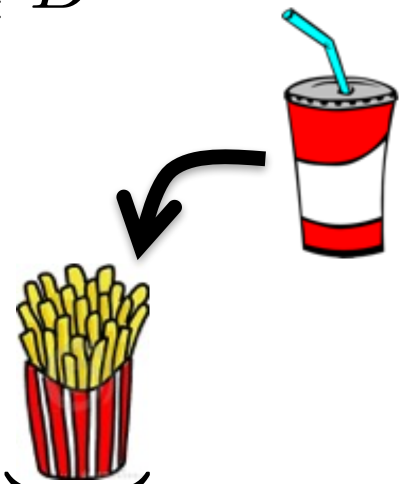
for all $A \subseteq B$
and s not in B



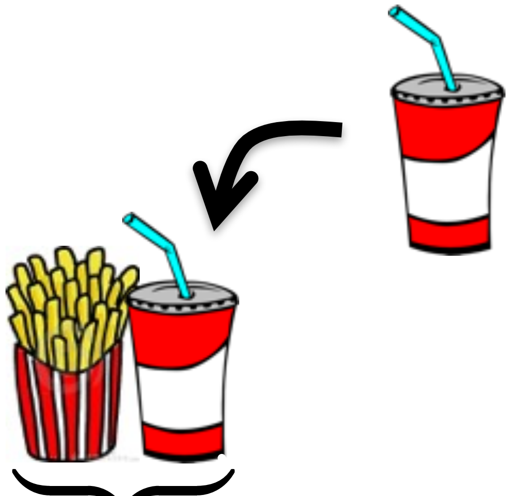
$$F(A \cup s) - F(A) \quad \geq \quad F(B \cup s) - F(B)$$

Diminishing costs: economies of scale

$A \subseteq B$



A



B

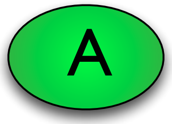
$$F(A \cup s) - F(A) \quad \geq \quad F(B \cup s) - F(B)$$

extra cost:
one drink

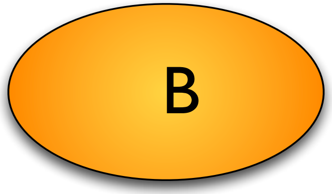
extra cost:
free refill ☺

Submodular set functions

- Diminishing gains: for all $A \subseteq B$



+ • e

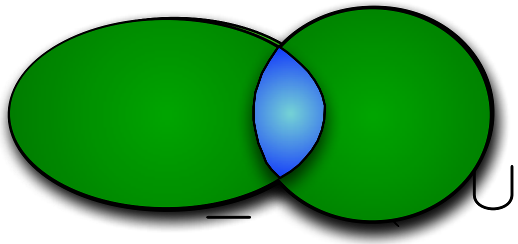


+ • e

$$F(A \cup e) - F(A) \geq F(B \cup e) - F(B)$$

- Union-Intersection: for all $S, T \subseteq \mathcal{V}$

$F(S) + F(T)$



$\cup T) + F(S \cap T)$

Example: modular function

each element $e \in \mathcal{V}$ has a weight $w(e)$

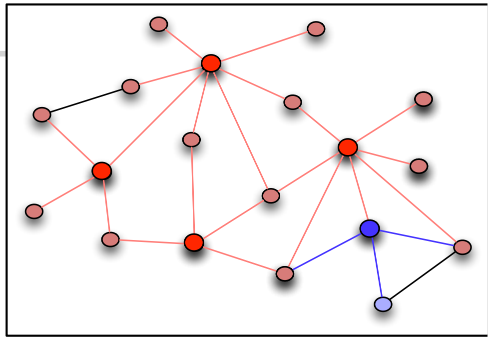
$$F(S) = \sum_{e \in S} w(e)$$

$A \subset B$

$$F(A \cup e) - F(A) = w(e) \quad = \quad F(B \cup e) - F(B) = w(e)$$

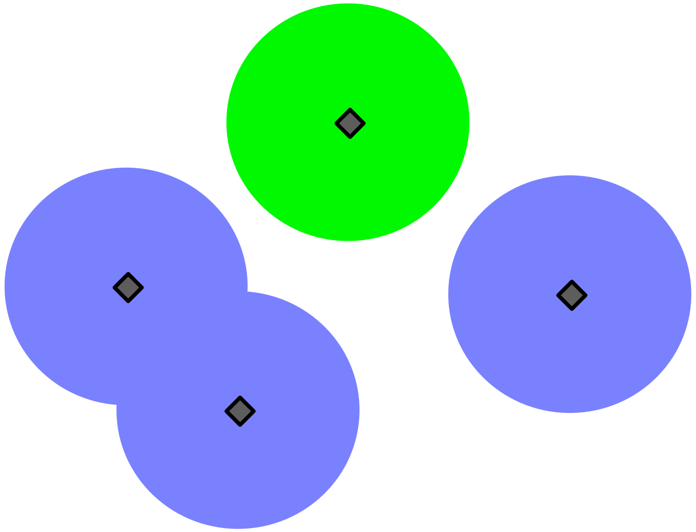
submodular *and* supermodular!

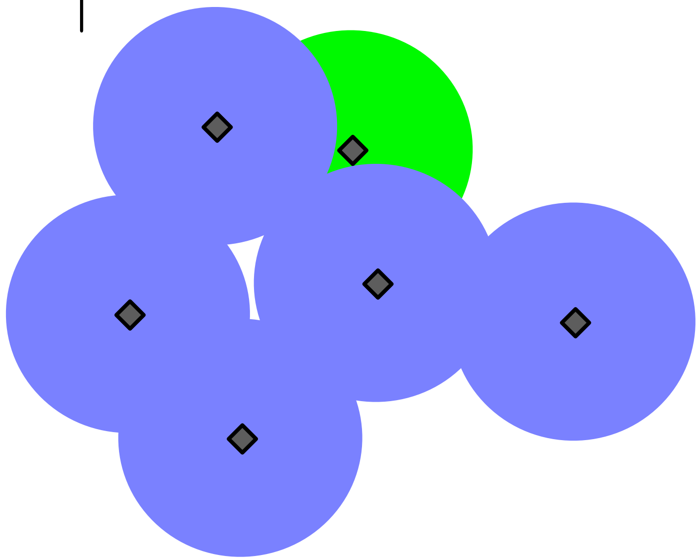
Example: coverage



$\mathcal{V}$ = all possible sensor locations

$$F(S) = \left| \bigcup_{v \in S} \mathrm{area}(v) \right|$$

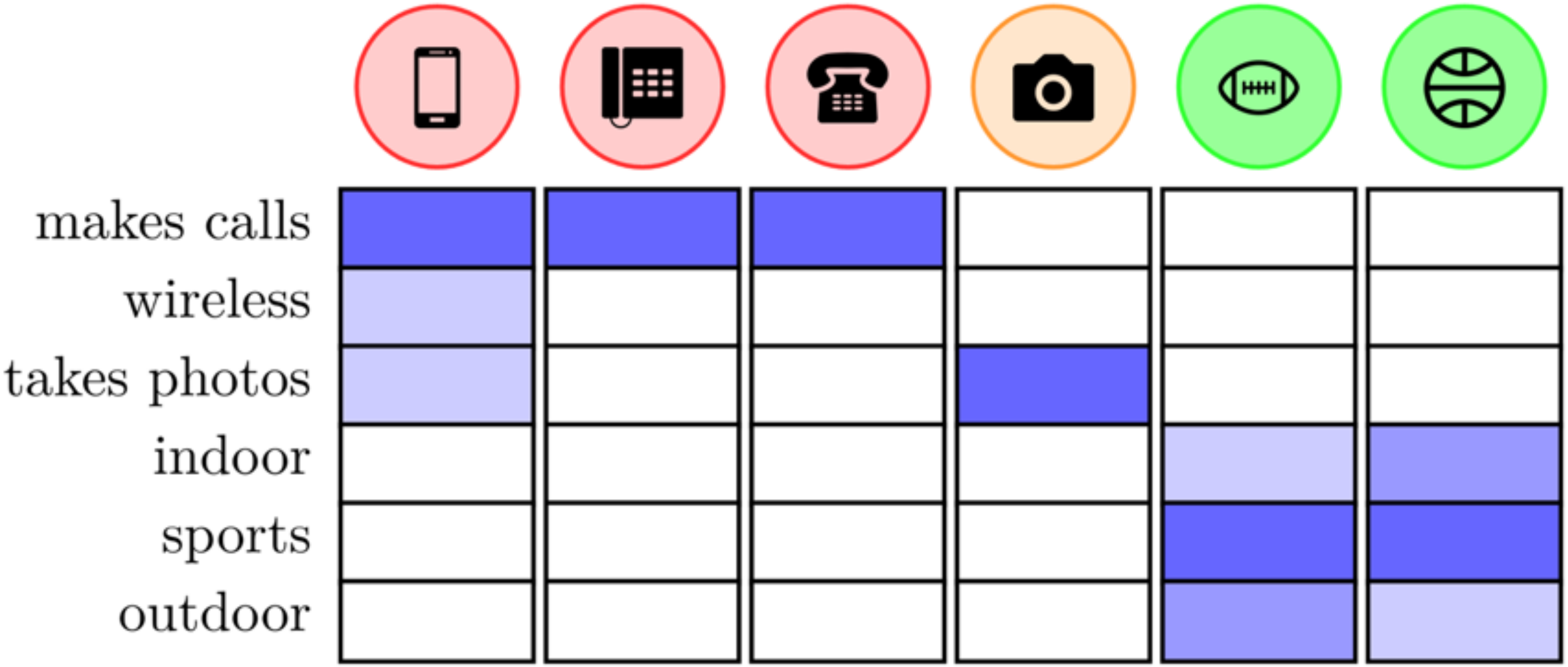




$$F(A \cup v) - F(A) \geq F(B \cup v) - F(B)$$

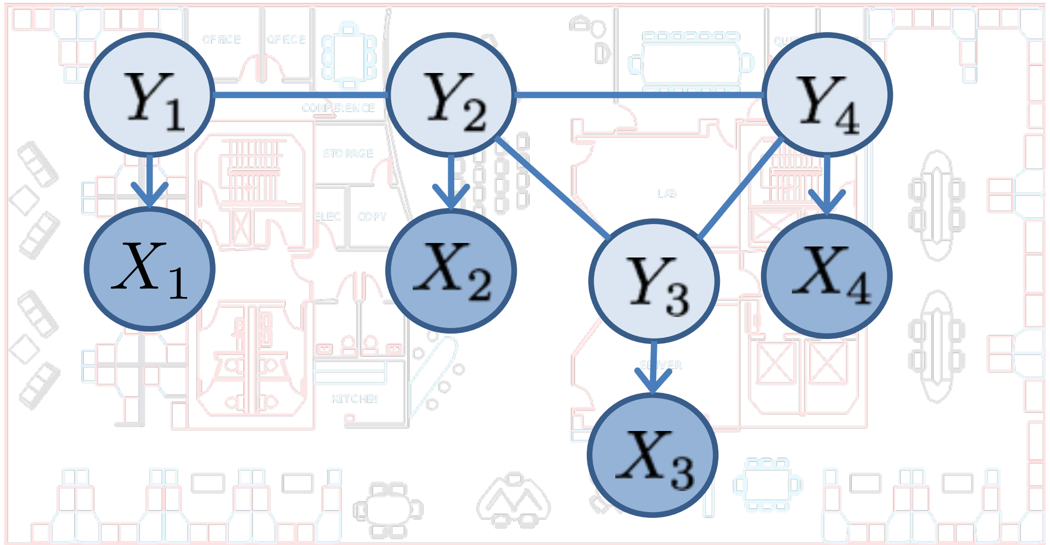
Example: Diversity in recommender systems (FLID)

[Tschiatschek, Djolonga, K, AISTATS 2016]



$$D(S) = \sum_{d=1}^{k} \left[\max_{i \in S} W_{i,d} \qquad\qquad\qquad \right]$$

Example: sensing



- $\mathcal{V}$ = random variables we can possibly observe
- Utility to have sensors in locations *A*:

$$F(A) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X}_A) = I(\mathbf{Y}; \mathbf{X}_A)$$

Uncertainty
about temperature Y
before sensing

Uncertainty
about temperature Y
after sensing

Example: entropy

$X_1, \ldots, X_n$ discrete random variables

$$F(S) = H(X_S) = \text{joint entropy of variables indexed by } S$$

$$A \subset B$$

$$
\begin{aligned}
H(X_{A\cup e}) - H(X_A) &= H(X_e | X_A) \\
&\geq H(X_e | X_B) \qquad \text{"information never hurts"} \\
&= H(X_{B\cup e}) - H(X_B)
\end{aligned}
$$

discrete entropy is submodular!

Submodularity and independence

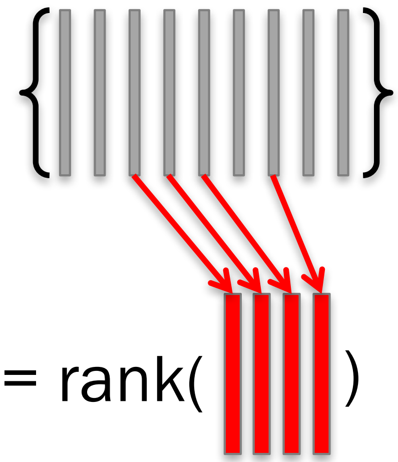
$X_1, \ldots, X_n$ discrete random variables

$X_i, i \in S$ statistically independent

⇔ H is modular/linear on S $\quad H(X_S) = \sum_{e \in S} H(X_e)$

Similarly: linear independence

$\mathcal{V} =$



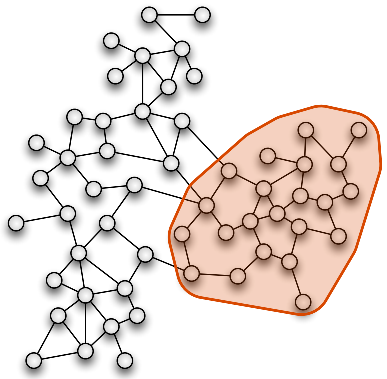
F(S) = rank( )

vectors in S linearly independent

⇔ F is modular/linear on S:

$F(S) = |S|$

Example: graph cuts

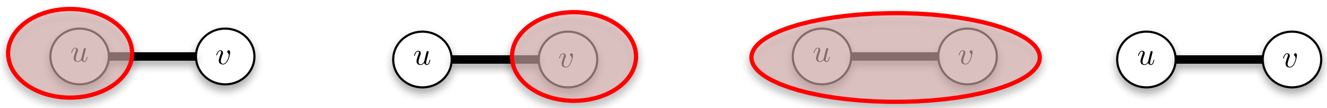


cut for one edge:

$$F(S) = \sum_{u \in S, v \notin S} w_{uv}$$

$$\underbrace{F(\{u\})}_{\color{blue}F(S)} + \underbrace{F(\{v\})}_{\color{blue}F(T)} \geq F(\{u,v\}) + F(\emptyset)$$

$\color{blue}F(S \cup T) + F(S \cap T)$



w_{uv} $\quad$ w_{uv} $\quad$ 0 $\quad$ 0

- cut of one edge is submodular!
- large graph: sum of edges

sum of submodular functions is submodular

Closedness properties

$F_1,\ldots,F_m$ submodular functions on V and $\lambda_1,\ldots,\lambda_m > 0$

Then: $F(A) = \sum_i \lambda_i F_i(A)$ is submodular

Submodularity closed under nonnegative linear combinations!

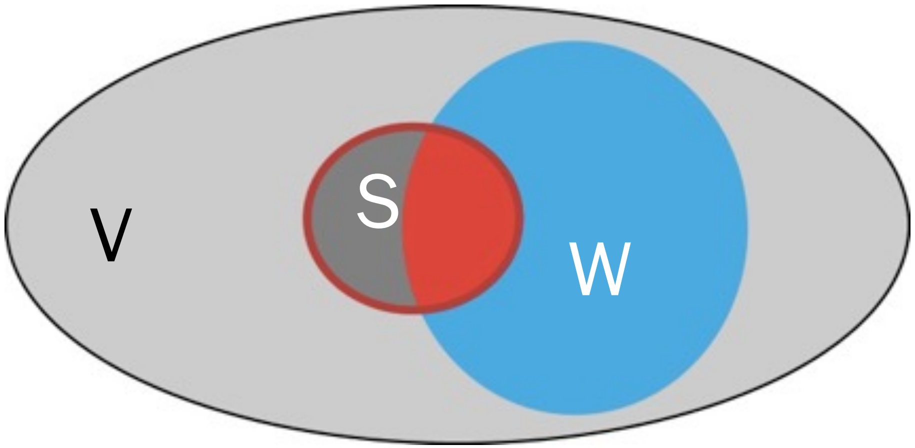
Extremely useful fact:

- $F_\theta(A)$ submodular ➔ $\sum_\theta P(\theta) F_\theta(A)$ submodular!
- Multicriterion optimization
- A basic proof technique! ☺

Other closedness properties

- Restriction: F(S) submodular on V, W subset of V

  Then $F'(S) = F(S \cap W)$ is submodular



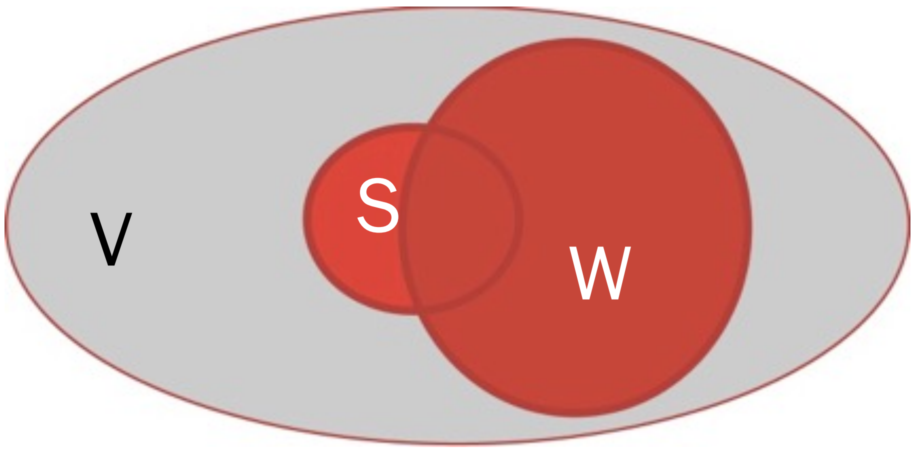
Other closedness properties

- Restriction: F(S) submodular on V, W subset of V

  Then $F'(S) = F(S \cap W)$ is submodular

- Conditioning: F(S) submodular on V, W subset of V

  Then $F'(S) = F(S \cup W)$ is submodular



Other closedness properties

- Restriction: F(S) submodular on V, W subset of V

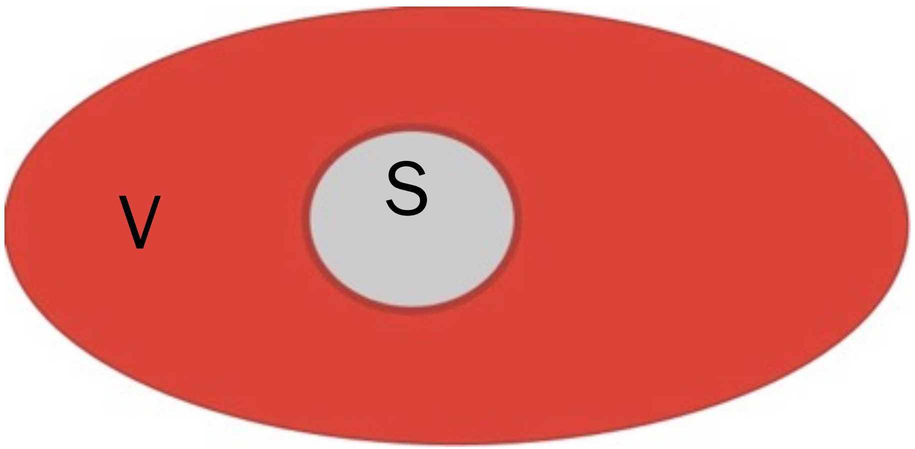
  Then $F'(S) = F(S \cap W)$ is submodular

- Conditioning: F(S) submodular on V, W subset of V

  Then $F'(S) = F(S \cup W)$ is submodular

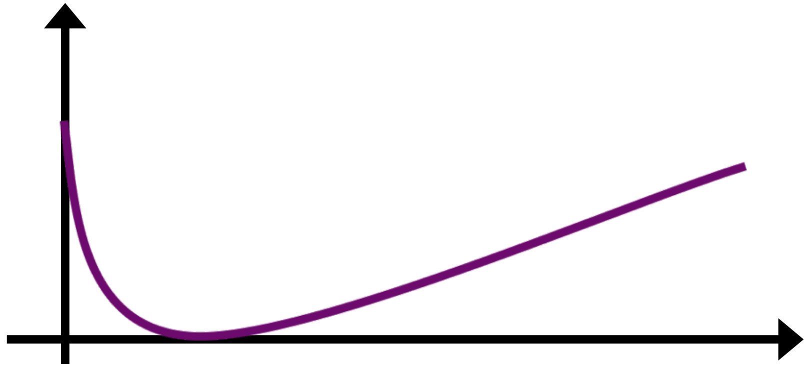
- Reflection: F(S) submodular on V

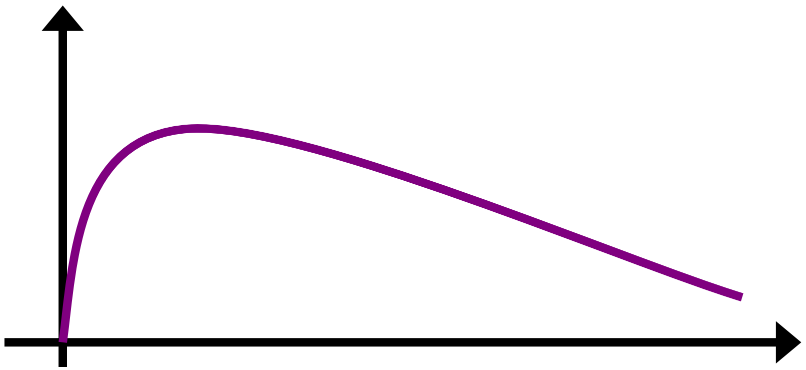
  Then $F'(S) = F(V \setminus S)$ is submodular



Submodularity ...

discrete convexity ....

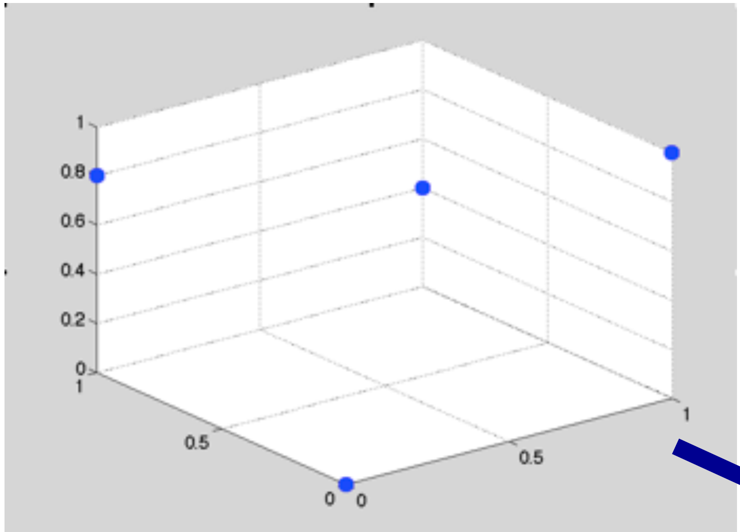


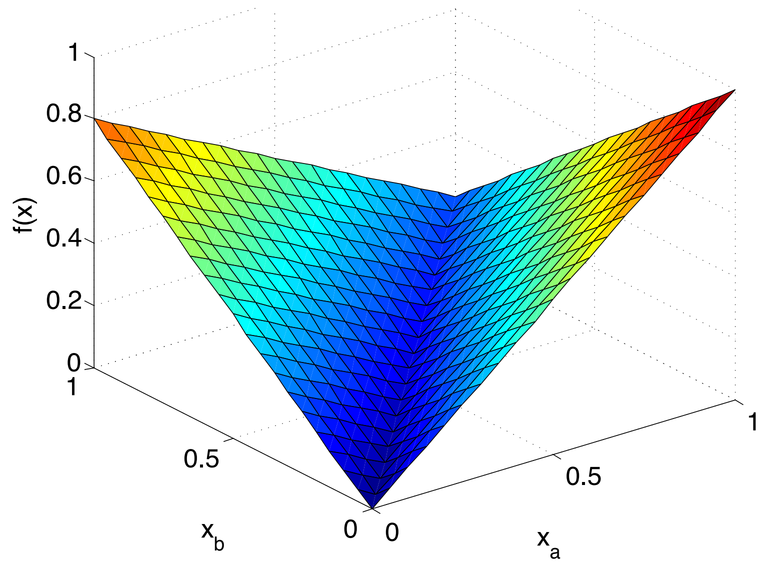


... or concavity?

Convex aspects

- convex extension
  - duality
  - efficient minimization





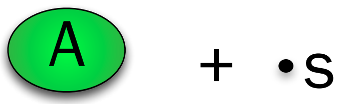
But this is only half of the story…

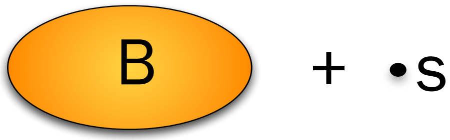
Concave aspects

- submodularity:

$$A \subseteq B, \quad s \notin B:$$

$$F(A \cup s) - F(A) \quad \geq \quad F(B \cup s) - F(B)$$

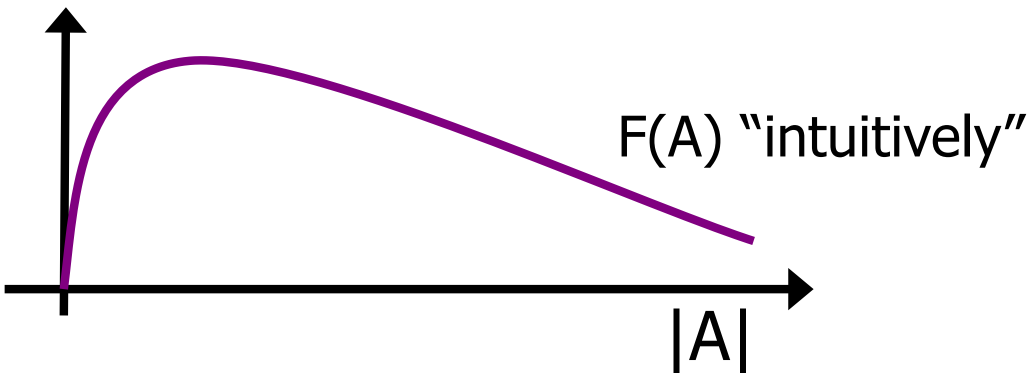




- concavity:

$$a \leq b, \quad s > 0:$$

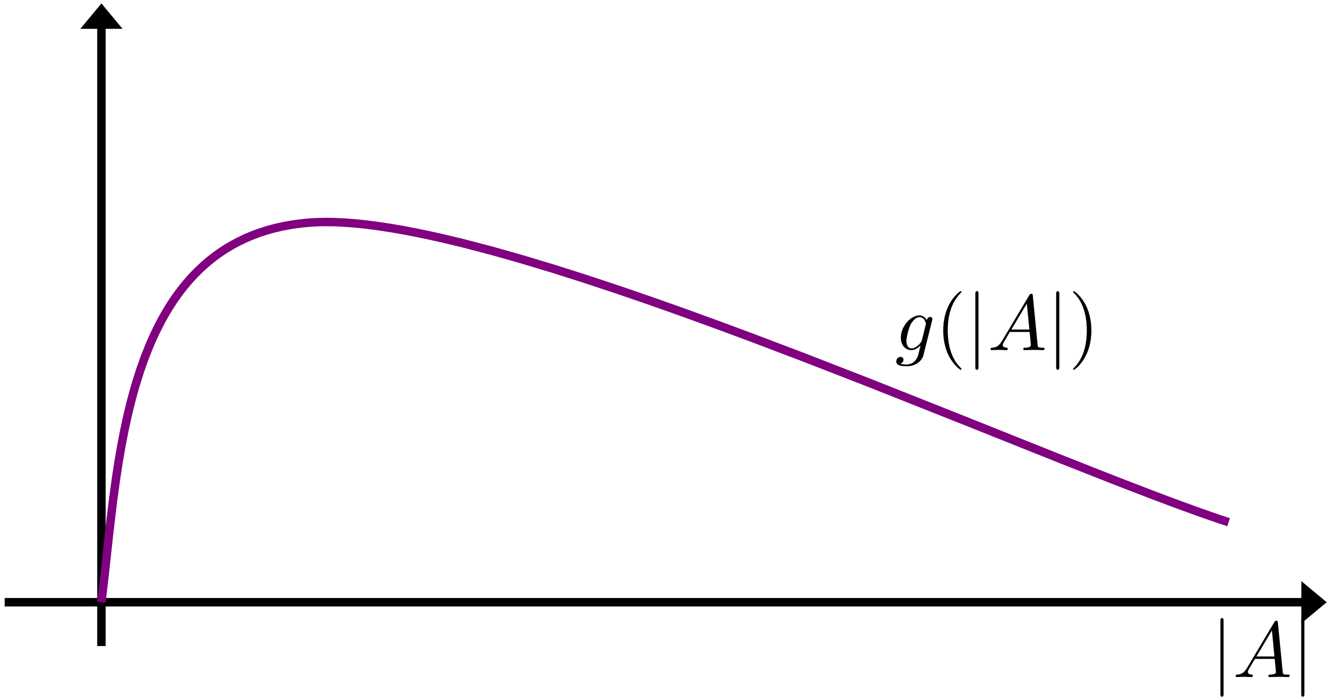
$$f(a+s) - f(a) \quad \geq \quad f(b+s) - f(b)$$



Submodularity and concavity

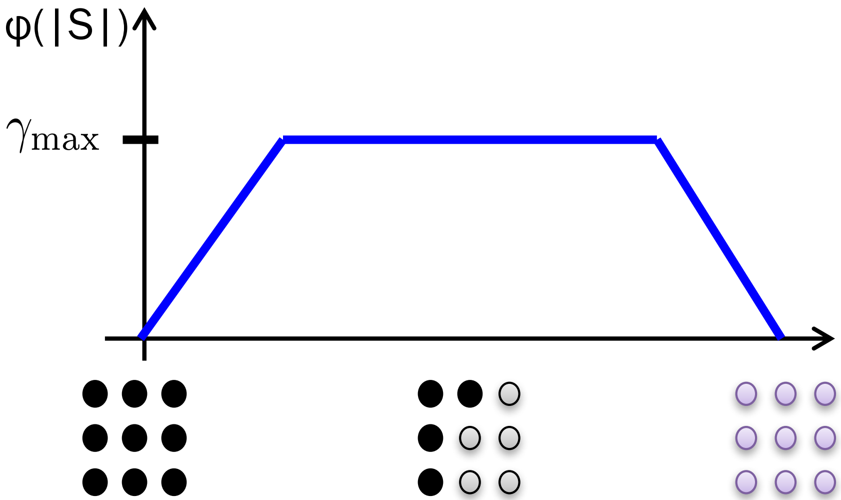
- suppose $g : \mathbb{N} \to \mathbb{R}$ and $F(A) = g(|A|)$

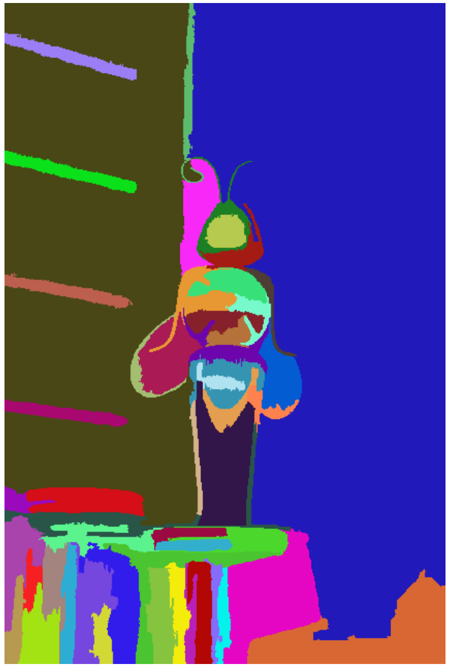
$F(A)$ submodular *if and only if* ...g is concave



Application: higher-order potentials

Pixels in a superpixel should have the same label

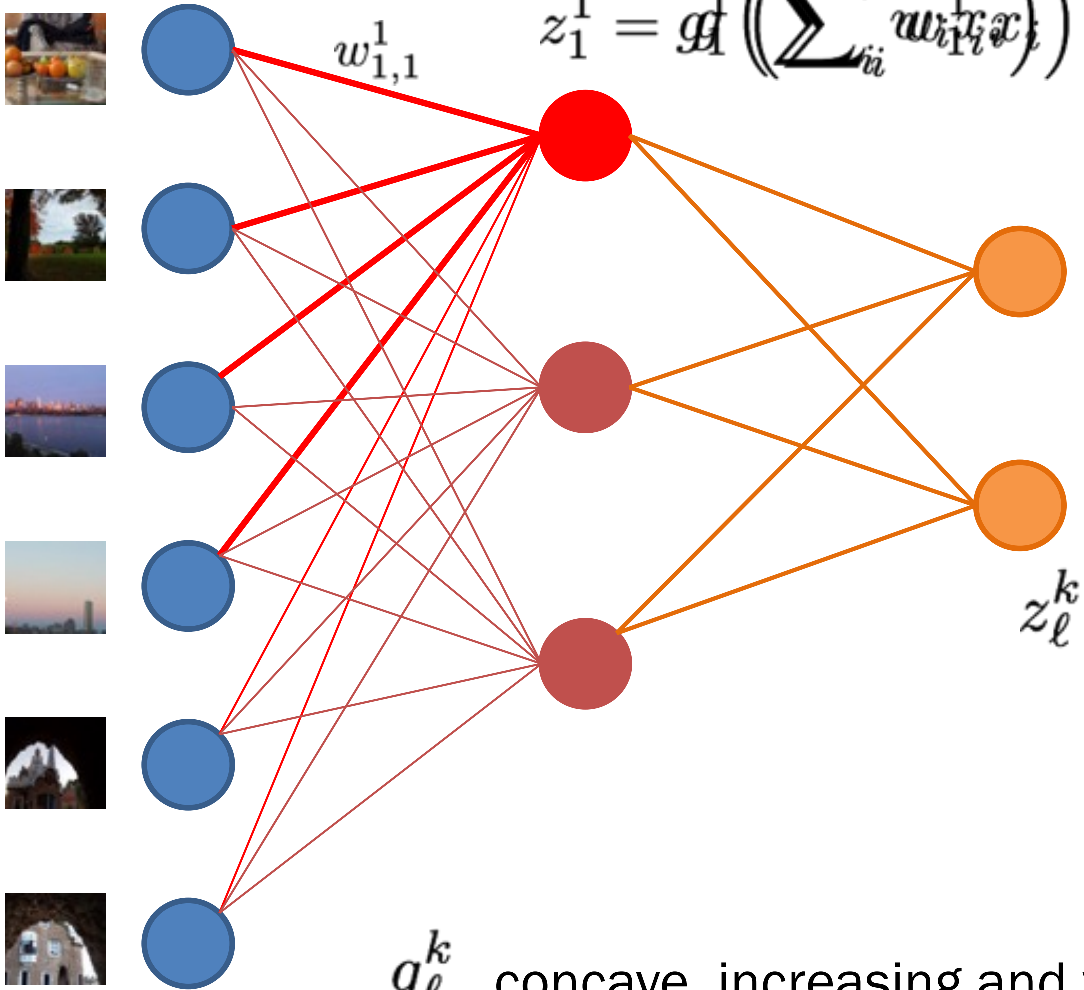




concave function of cardinality ➔ submodular ☺

Deep Submodular Functions

$$z_1^1 = g_1^1\left(\sum_i w_{1i}^1 x_i\right)$$



$$F(S) = \sum_\ell z_\ell^K$$

$$z_\ell^k = g_\ell^k\left(\sum_j w_{\ell,j}^k z_j^{k-1}\right)$$

g_ℓ^k concave, increasing and weights nonnegative

$x \in \{0,1\}^n$

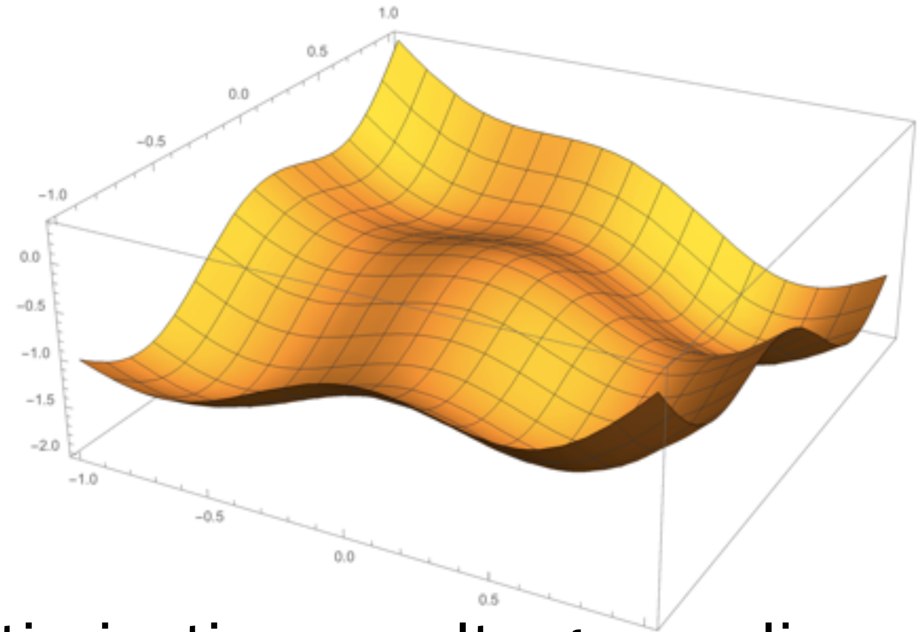
(Dolhansky & Bilmes 16)

Submodularity more generally

- Lattices and continuous functions

$$f(x) + f(y) \;\geq\; f(x \vee y) + f(x \wedge y)$$

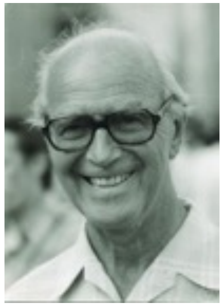
subclass: diminishing returns (DR) – submodular fn's



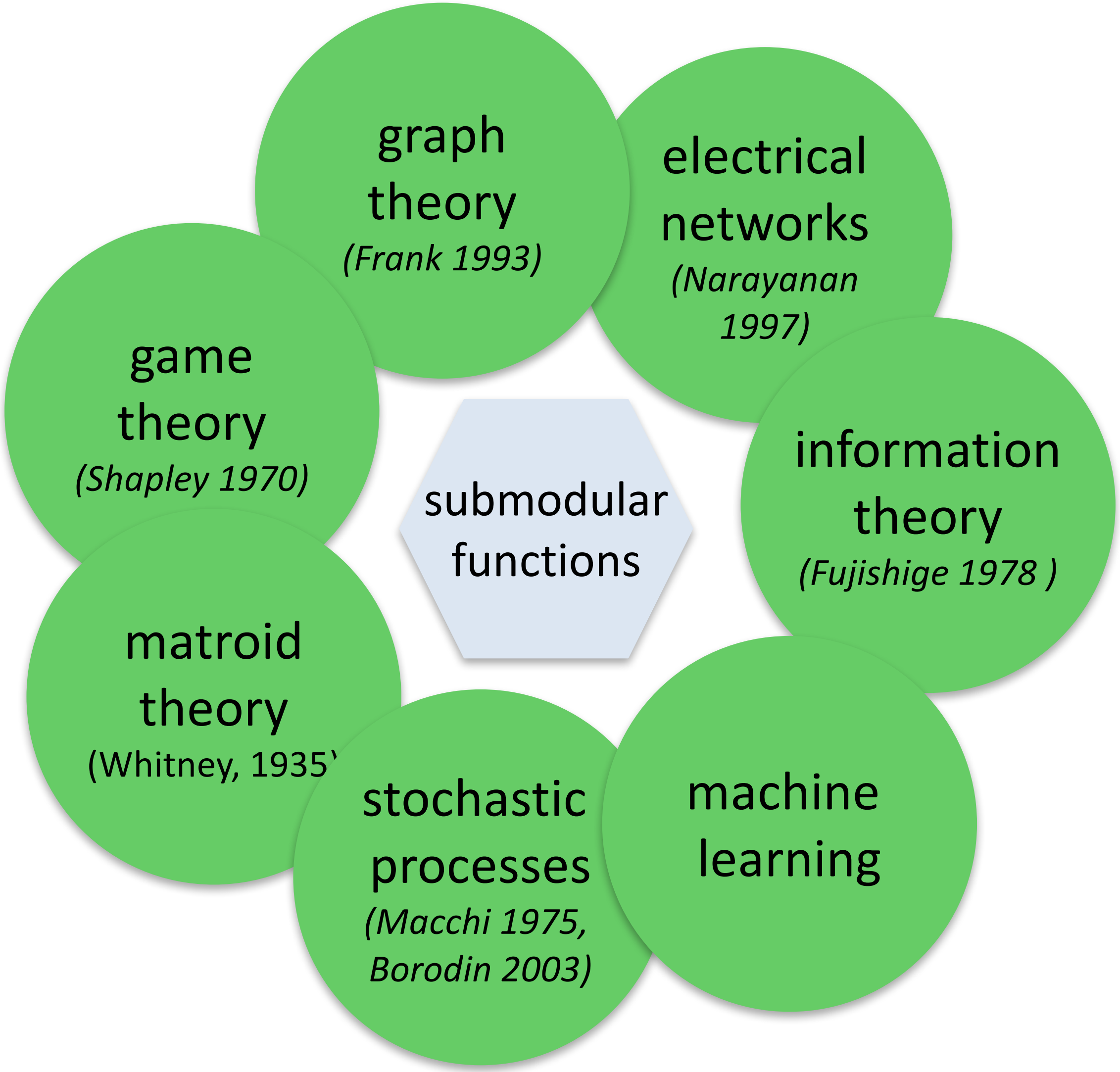
- Many optimization results generalize

(Milgrom-Shannon 94; Topkis 98; Murota 03; Kapralov-Post-Vondrak 10; Soma et al 2014-16; Bach 2015; Ene & Nguyen 2016; Bian-Mirzasoleiman-Buhmann-Krause 16)

Origins and history



G. Choquet



J. Edmonds

L.S. Shapley

L. Lovász

nonconvex optimization
lattice / continuous submodularity
many optimization & duality results generalize

probability measures
log-supermodular (→positive assoc.)
log-submodular (←negative assoc.)
sampling, mode,
approx. partition function

submodular set functions

convexity:
minimization
max. coherence

dim. returns:
maximization
max. diversity

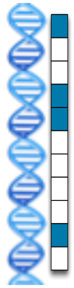
many examples:

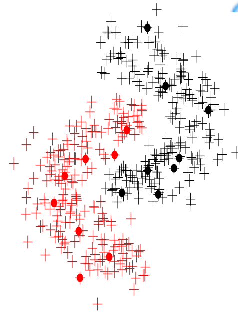
- linear/modular functions
- entropy
- mutual information
- rank functions
- coverage
- diffusion in networks
- volume
- graph cut …

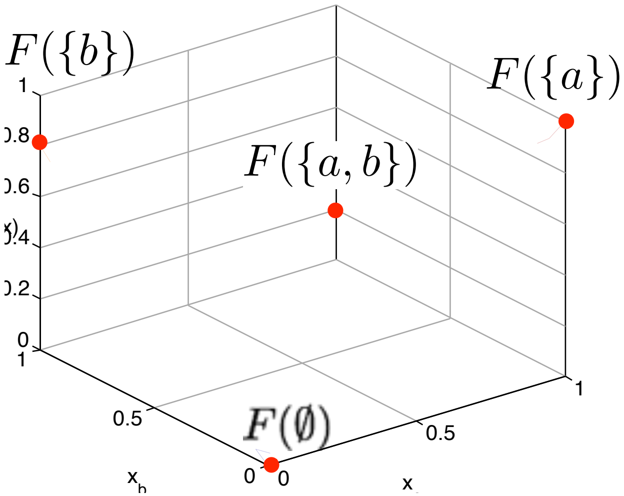
Submodular minimization

$$\min_{S \subseteq \mathcal{V}} F(S)$$

"maximize coherence"







Idea: relaxation

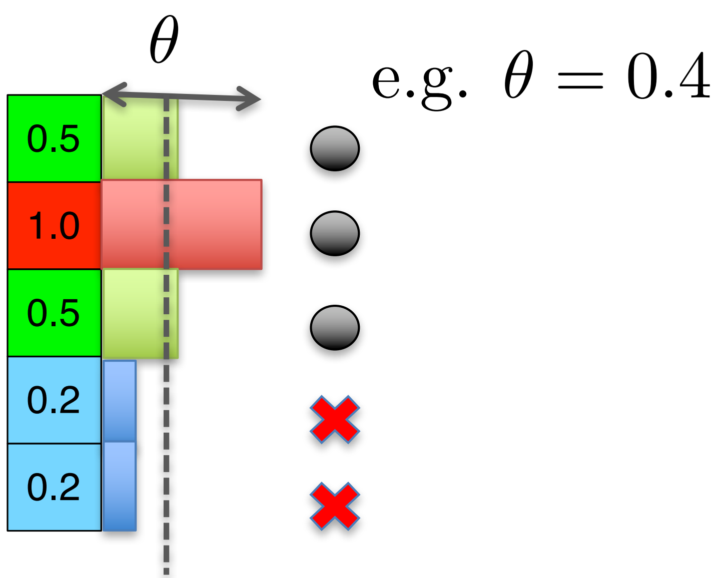
$$\min_{x \in \{0,1\}^n} F(x) \longrightarrow \min_{x \in [0,1]^n} f(x)$$

Lovász extension

- expectation:

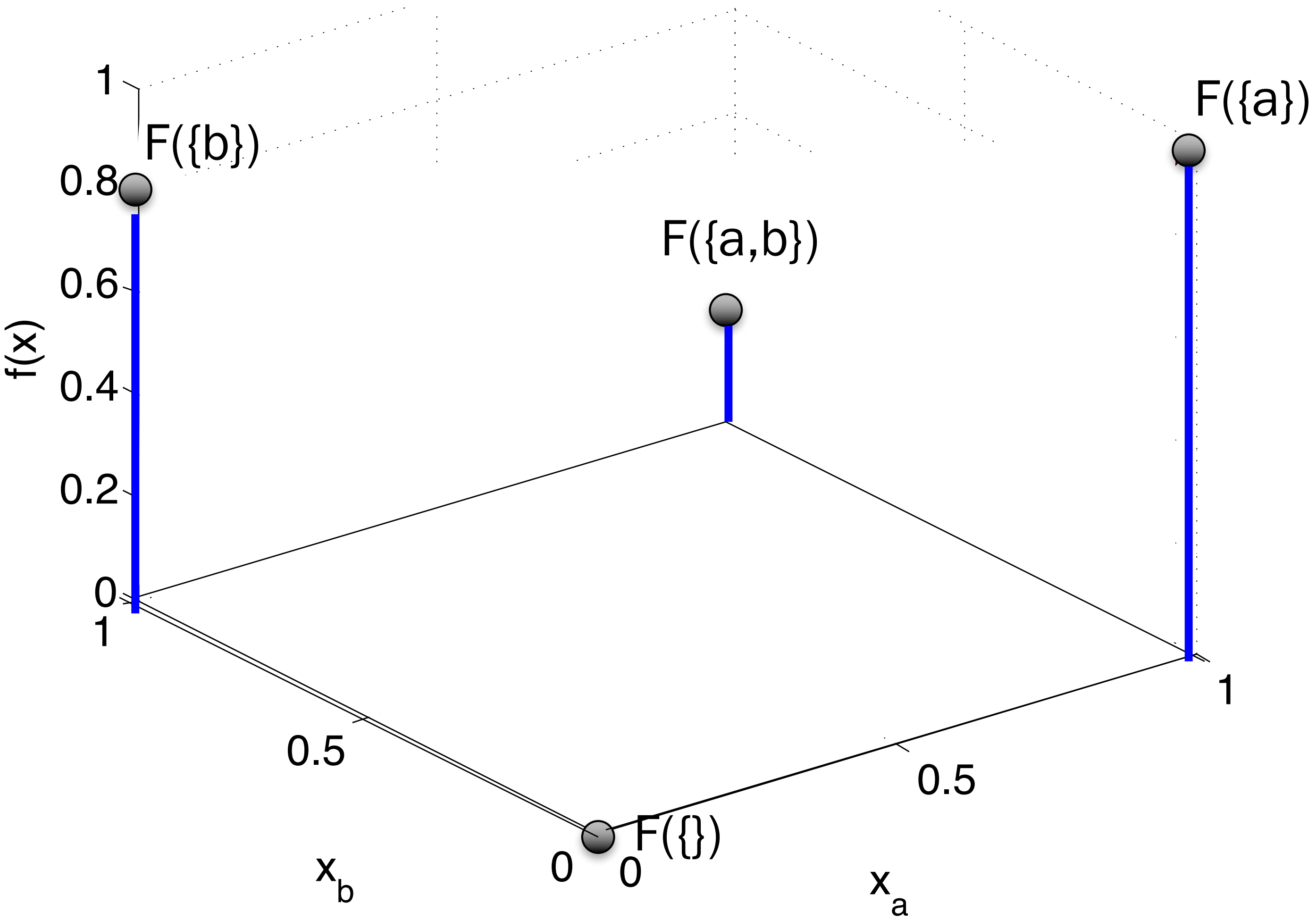
$$f(x) = \mathbb{E}_{\theta \sim x}[F(S_\theta)]$$

- sample threshold $\theta \in [0, 1]$ uniformly
- $S_\theta = \{e \mid x_e \geq \theta\}$



Lovász extension: example

$$f(x) = \mathbb{E}_\theta[F(S_\theta)]$$



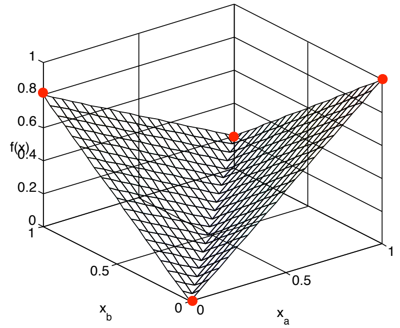
A	F(A)
{}	0
{a}	1
{b}	.8
{a,b}	.2

Alternative characterization

$$f(x) = \mathbb{E}_{\theta \sim x}[F(S_\theta)]$$

if F is submodular, this is equivalent to:

$$\boxed{f(x) = \max_{y \in \mathcal{B}_F} \; y^\top x}$$



Theorem *(Edmonds 1971, Lovász 1983)*

Lovász extension is convex $\Leftrightarrow$ F is submodular.

ETH

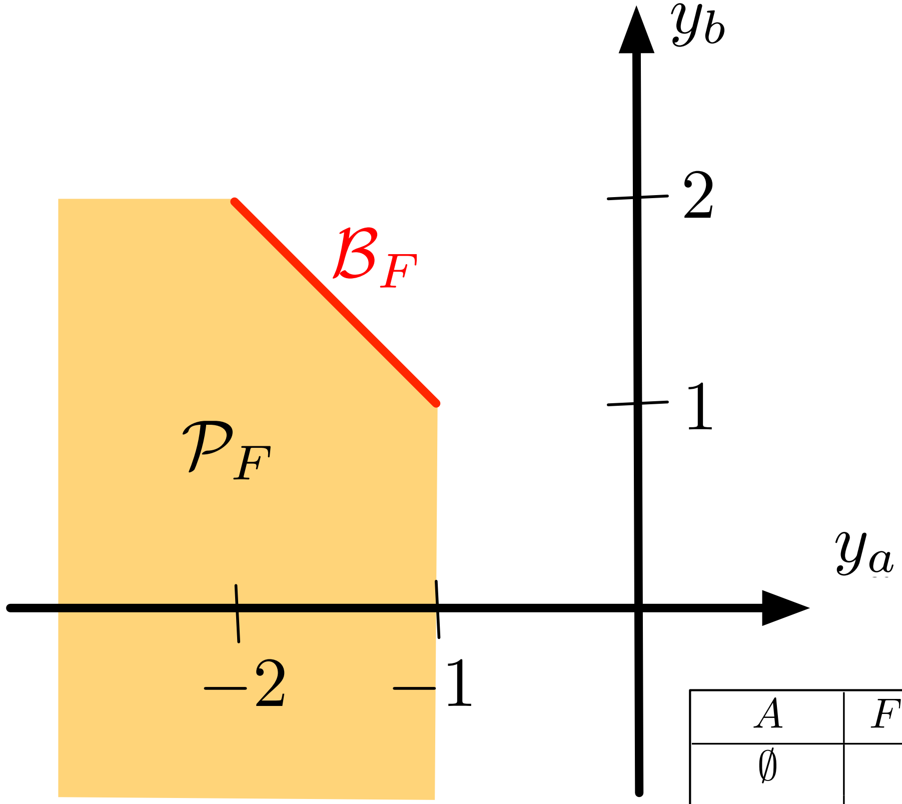
Submodular polyhedra

submodular polyhedron:

$$\mathcal{P}_F = \left\{ y \in \mathbb{R}^n \mid \sum_{a \in A} y_a \le F(A) \text{ for all } A \subseteq \mathcal{V} \right\}$$

base polytope:

$$\mathcal{B}_F = \left\{ y \in \mathcal{P}_F \mid \sum_{a \in \mathcal{V}} y_a = F(\mathcal{V}) \right\}$$



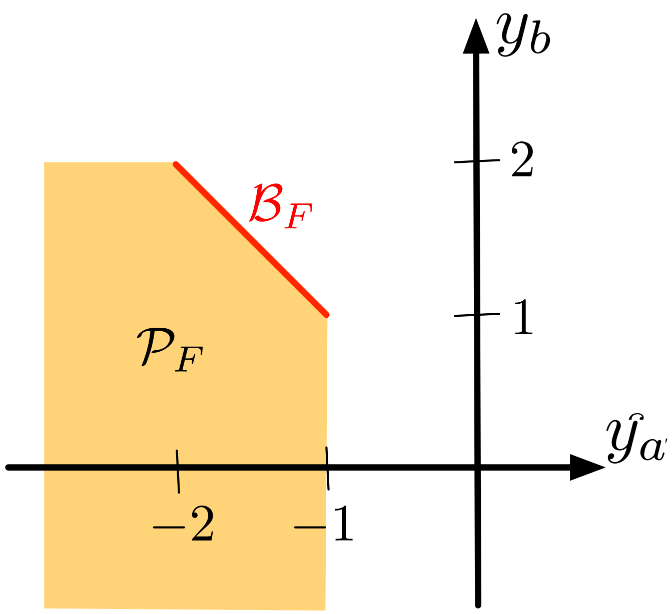
A	$F(A)$
$\emptyset$	0
a	-1
b	2
$\{a,b\}$	0

Examples:

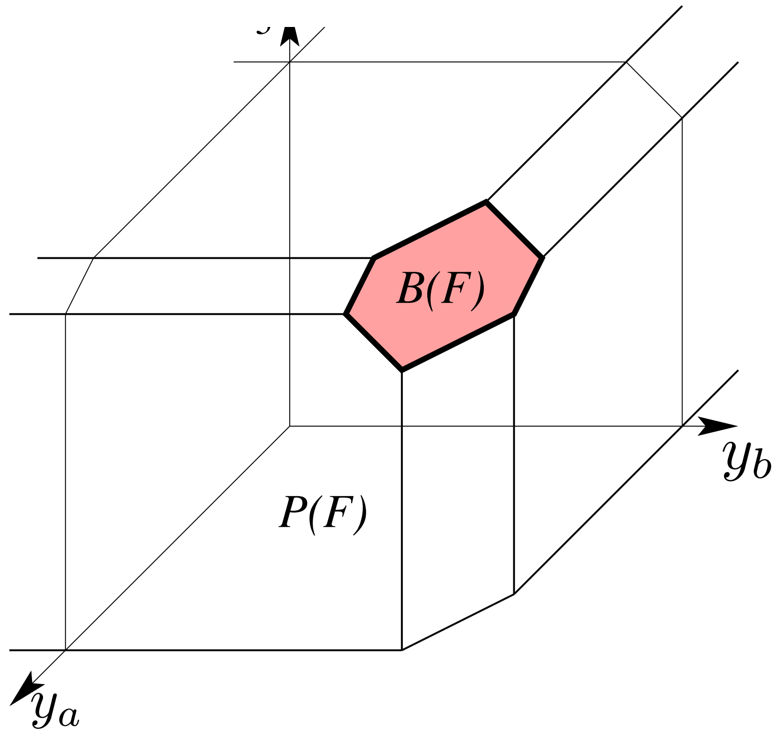
- probability simplex
- spanning tree polytope
- permutahedron

Base polytopes

2D (2 elements)



3D (3 elements)



The magic of base polytopes

$$f(x) = \max_{y \in \mathcal{B}_F} \; y^\top x \qquad = \max_{y \in \mathcal{B}_F} \sum_i y_i x_i$$

- Linear optimization over the base polytope?
  exponentially many constraints (one for each subset)

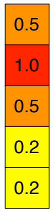
- Edmonds 1971: greedy works ☺

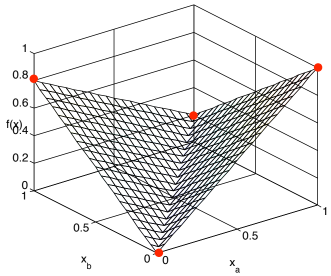
1. sort cost vector $x_{\pi(1)} \geq x_{\pi(2)} \geq \dots$
2. gives sets $S_i = \{\pi(1), \dots, \pi(i)\}$
3. Set $y_{\pi(i)} = F(S_i) - F(S_{i-1})$

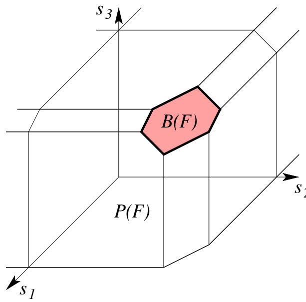
Implications: can compute

- Lovász extension
- subgradients of Lovász extension

Putting things together







$$\min_{S \subseteq \mathcal{V}} F(S) \quad = \quad \min_{x \in \{0,1\}^n} F(x) \quad \longrightarrow \quad \min_{x \in [0,1]^n} f(x)$$

1. relaxation: convex optimization
   computable subgradients

   $\longleftarrow$ many ways to do Step 1

2. relaxation is exact!
   pick elements with positive coordinates $S^* = \{e \mid x_e^* > 0\}$

➔ submodular minimization in polynomial time!
(Grötschel, Lovász, Schrijver 1981)

Submodular minimization

convex optimization

- ellipsoid method
  (Grötschel-Lovasz-Schrijver 81)
- subgradient method …
  (…, Chakrabarty-Lee-Sidford-Wong 16)
- minimum-norm point / Fujishige-Wolfe algorithm
  (different relaxation)
  (Fujishige-Isotani 11)
- …

combinatorial methods

- first polynomial-time:
  (Schrijver 00, Iwata-Fleischer-Fujishige-01)
  …
- $O(n^4 T + n^5 \log M)$ *(Iwata 03)*
  $O(n^6 + n^5 T)$ *(Orlin 09)*

Latest:

$$O(n^2 T \log nM + n^3 \log^c nM)$$

$$O(n^3 T \log^2 n + n^4 \log^c n)$$

(Lee-Sidford-Wong 15)

Different relaxation

$$\min_x \; f(x) + \frac{1}{2}\|x\|^2$$

solves

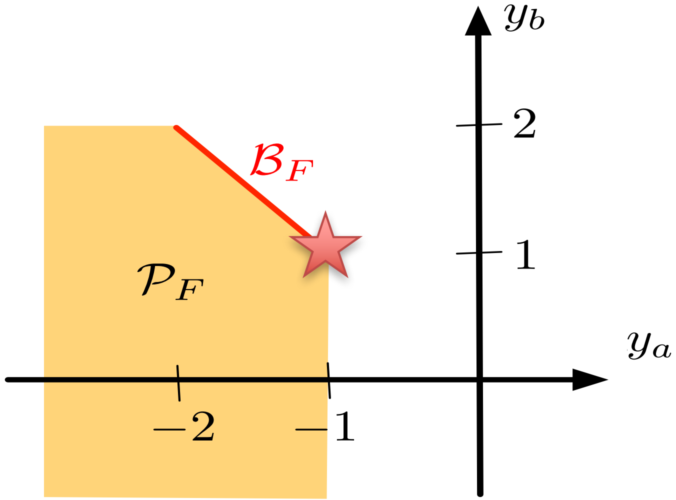
$$\min_{S \subseteq \mathcal{V}} F(S) + \alpha|S|$$ for all α

threshold optimal solution x^* at α

- dual problem: minimum norm point of base polytope

$$\min_{y \in \mathcal{B}_F} \|y\|^2$$

- optimization:
  Frank-Wolfe,
  Wolfe's algorithm

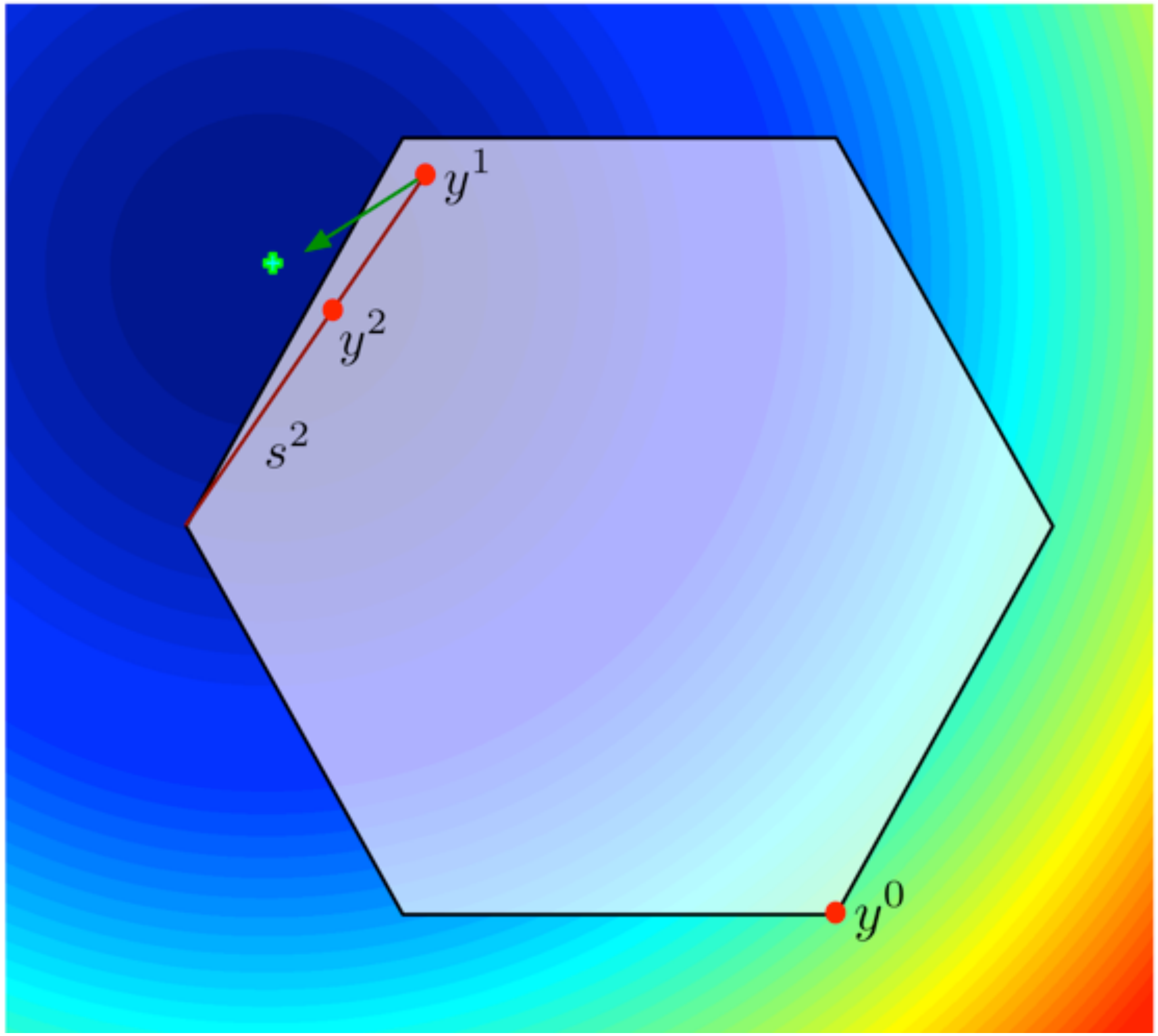


A	$F(A)$
$\emptyset$	0
a	-1
b	2
$\{a,b\}$	0

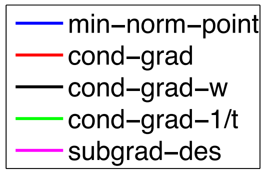
Minimum norm point

$$\min_{y \in \mathcal{B}_F} \|y\|^2$$

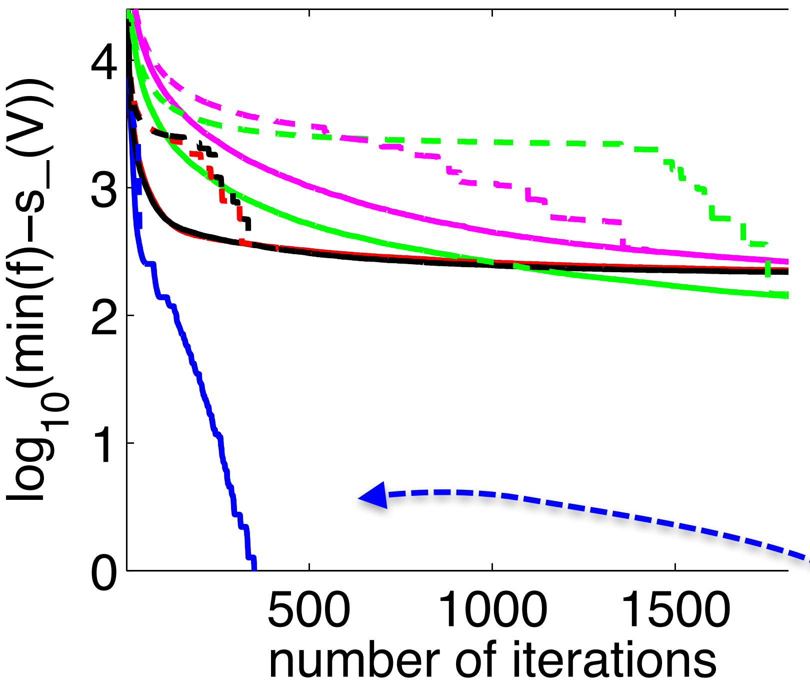
$$s^t \in \arg\max_{s \in \mathcal{B}_f} \langle -\nabla g(y^t), s \rangle$$



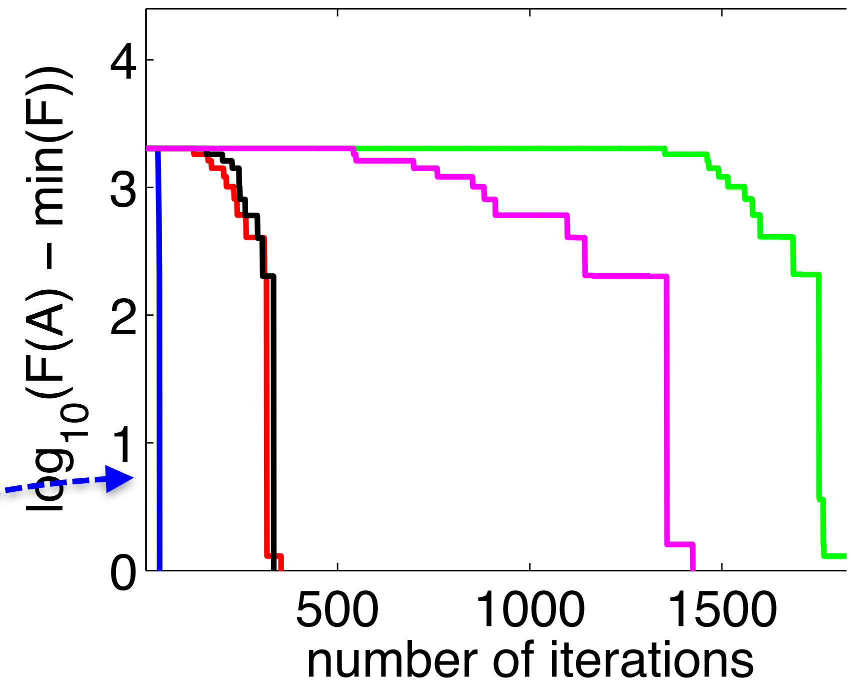
Empirically



convergence of relaxation



convergence of S



min-norm point

(Figure from Bach, 2012)

Submodularity and convexity

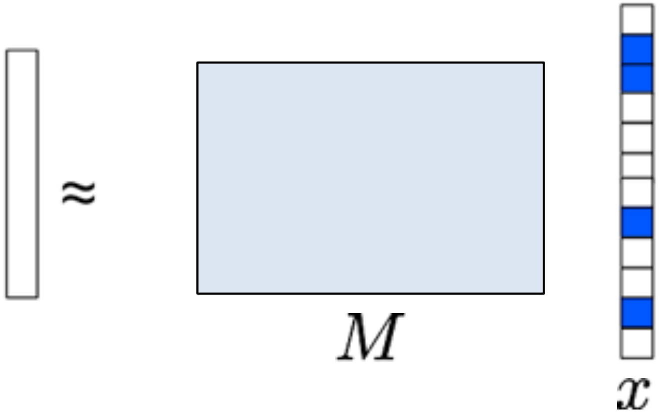
- convex Lovász extension
  - easy to compute: greedy algorithm (special polyhedra!)

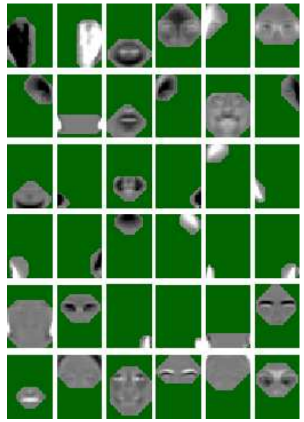
- submodular minimization via convex optimization: exact
- duality results

- structured sparsity *(Bach 10)*
- decomposition & parallel algorithms
  (Komodakis-Paragios-Tziritas 11, Stobbe-Krause 10, Jegelka-Bach-Sra 13, Nishihara-Jegelka-Jordan 14, Ene-Nguyen 15)
- variational inference *(Djolonga-Krause 14)*
- …

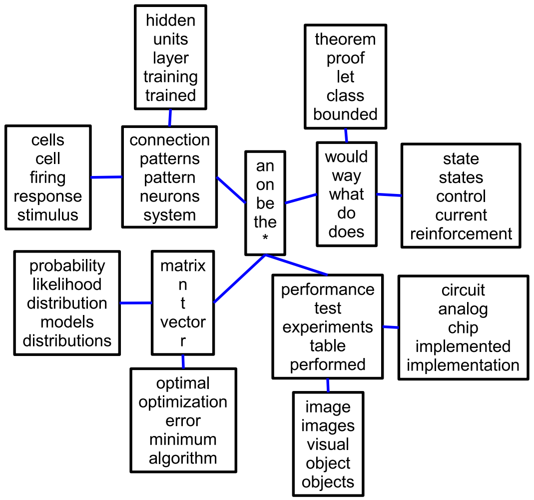
Structured sparsity and submodularity

$$y = M{\color{red}x} + \text{noise}$$





Structured sparse PCA

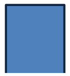


$$\min_{x} \; \|y - Mx\|^2 \; + \lambda\Omega(x)$$

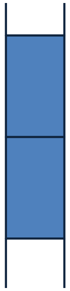
Sparsity

$$\min_x \; \|y - Mx\|^2 \; + \lambda \Omega(x)$$

- x sparse
- x structured sparse



discrete regularization on support S of x

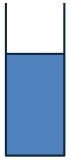


submodular function

$$\Omega(x) = \|x\|_0 \quad = |S|$$

$$\Omega(x) = F(S)$$

relax to convex envelope



➔ Lovász extension

$$\Omega(x) = \|x\|_1$$

$$\Omega(x) = f(|x|)$$

Optimization: submodular minimization (min-norm)

(Bach2010)

Submodular min: special cases

- "Graph-representable": reduction to minimum cut
  (Billionet & Minoux 85, Kolmogorov-Zabih 04, Freedman & Drineas 05, Živný 09, Živný & Jeavons 10, ...)

- Decomposable functions $F(S) = \sum_i F_i(S)$
  (Stobbe-Krause 10, Komodakis-Paragios-Tziritas 11, Kolmogorov 12, Jegelka-Bach-Sra 13, Nishihara-Jegelka-Jordan 14, Ene-Nguyen 15, Fix-Joachims-Park-Zabih 13, Fix-Wang-Zabih 14)

- Symmetric functions $F(S) = F(\mathcal{V} \setminus S)$
  (Queyranne 98) $O(n^3)$

nonconvex optimization
lattice / continuous submodularity
many optimization & duality results generalize

probability measures
log-supermodular (→positive assoc.)
log-submodular (←negative assoc.)
sampling, mode,
approx. partition function

submodular set functions

convexity:	dim. returns:
minimization	maximization
max. coherence	*max. diversity*

many examples:

- linear/modular functions
- entropy
- mutual information
- rank functions
- coverage
- diffusion in networks
- volume
- graph cut …

Outline

1. What is Submodularity?
   Examples, connections

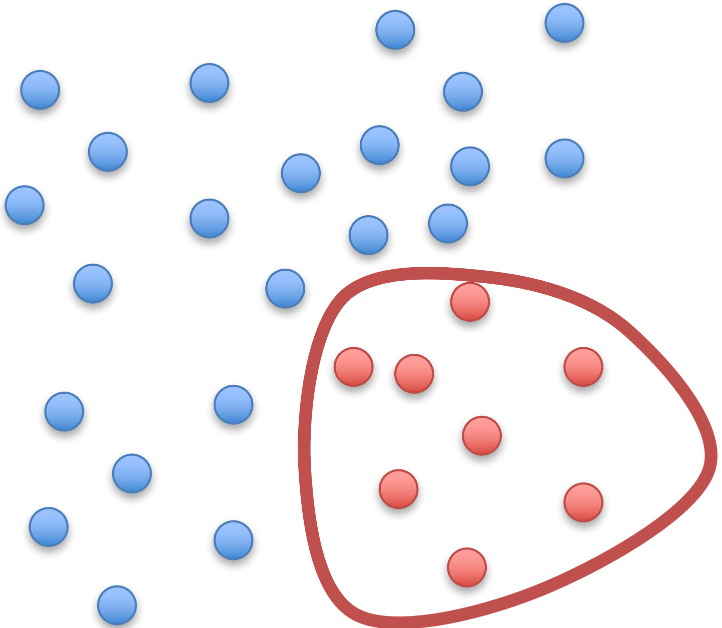
2. Submodular minimization

3. Submodular maximization

TOMORROW

4. Advanced Topics
   submodularity in deep learning, probabilistic inference, active learning, bandits, ...

Submodular Maximization



$S \subseteq \mathcal{V}$

- ground set $\mathcal{V}$
- submodular function

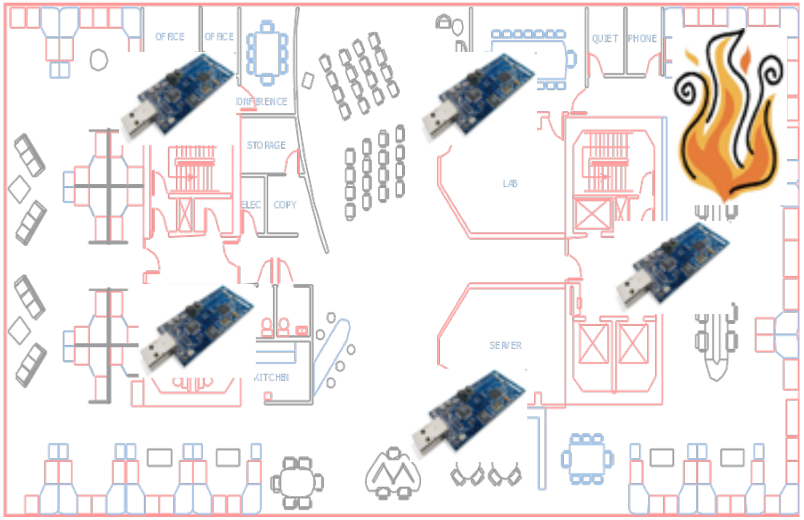
$$F : 2^{\mathcal{V}} \rightarrow \mathbb{R}$$

$$\boxed{\max \quad F(S)}$$

Often s.t. to some constraints

Survey: Krause & Golovin (2014) "Submodular Function Maximization"

Application: Information Gathering

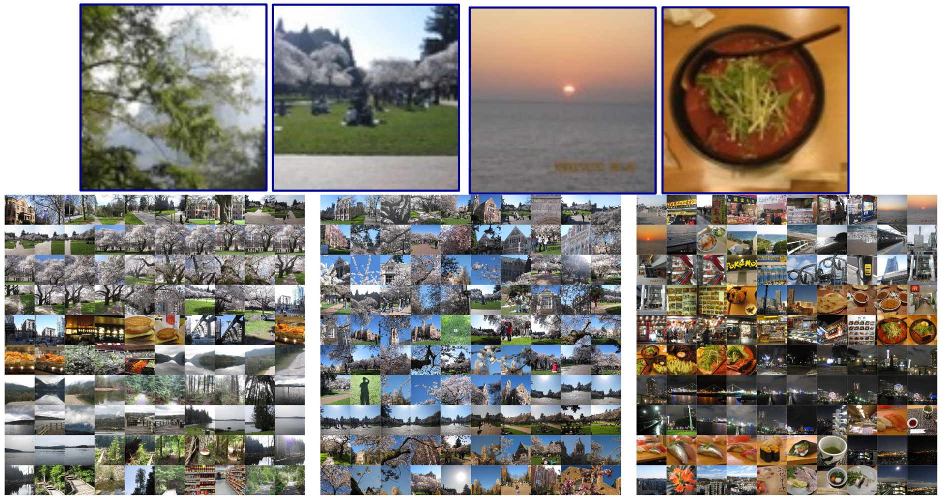


- where put sensors?
- which experiments?
- which labels?

$$F(S) = \text{“information”}$$

(Krause & Guestrin ‘05, Hoi-Jin-Zhu-Lyu ’06, Das & Kempe ’08, …)

Application: Data Summarization

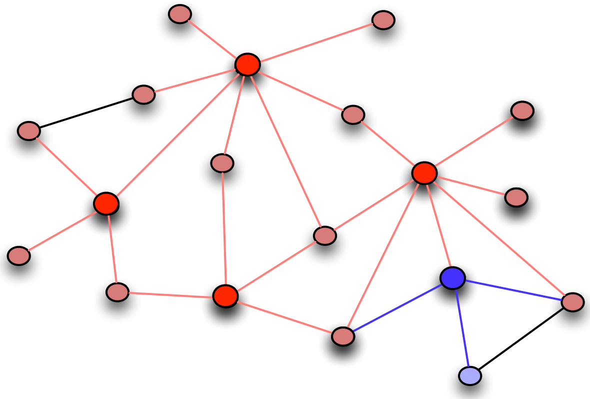


- which text, images, videos?
- which data points for training?

F(S) = “relevance, diversity, …”

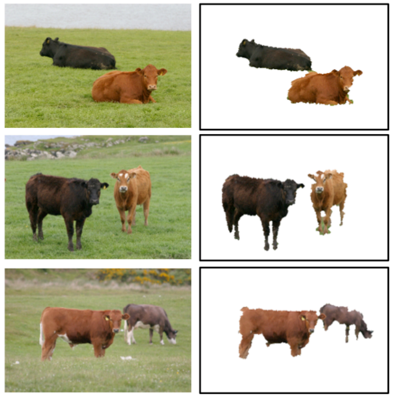
(El-Arini et al ‘09, Yue & Guestrin ‘09, Gomes & Krause’10, Lin & Bilmes ‘11, …)

More maximization ...



Influence maximization
(Kempe, Kleinberg, Tardos '03)

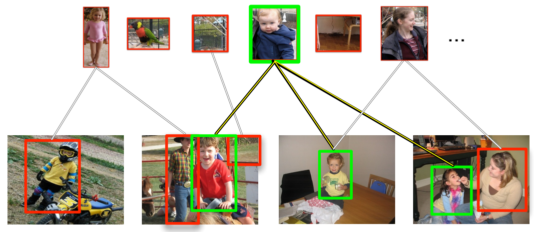
co-segmentation
by maximizing
anisotropic diffusion
(Kim-Xing-Fei Fei-Kanade '11)

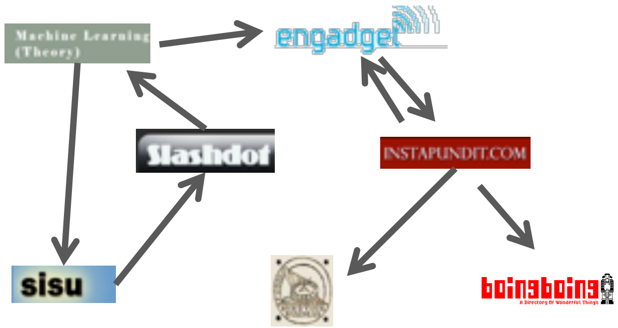


$$\max \quad F(S)$$

diverse
recommendations
(Yue & Guestrin '11)

weakly supervised
object detection
(Song-Girshick-Jegelka-Mairal-Harchaoui-Darrell '14)

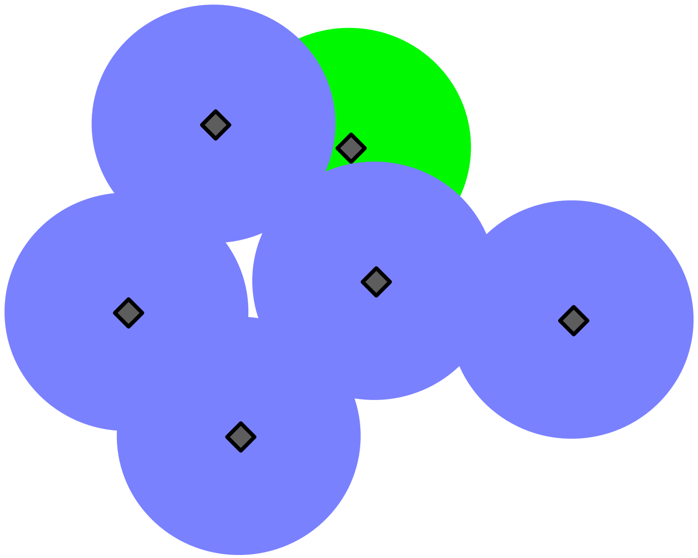


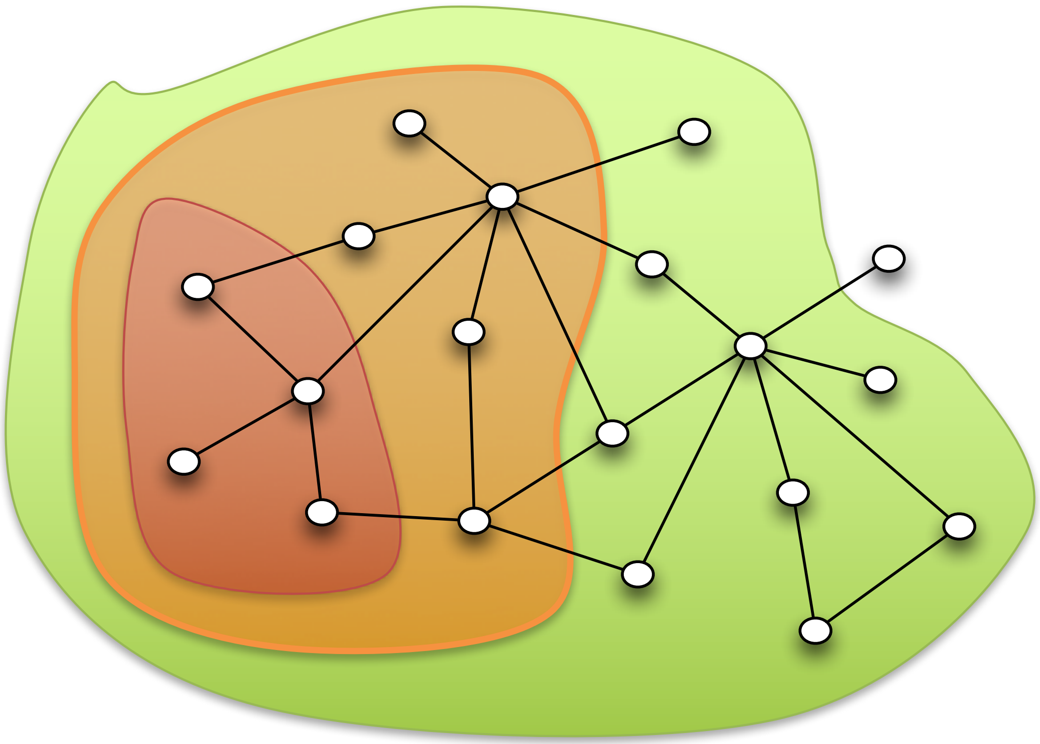


inferring networks
(Gomez Rodriguez et al 2012)

Monotonicity

$$\text{if} \quad S \subseteq T \quad \text{then} \quad F(S) \leq F(T)$$





3 5 1

Maximizing monotone functions

$$\text{if } A \subseteq B \quad \text{then} \quad F(A) \leq F(B)$$

$$\boxed{\max \ F(S)}$$

- NP-hard
- Approximation: Greedy algorithms

Maximizing monotone functions

$$\max_S \; F(S) \;\text{ s.t. }\; |S| \leq k$$

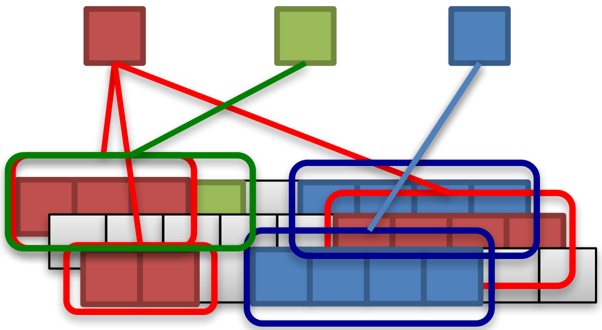
Greedy algorithm:

$S_0 = \emptyset$

for *i* = *0, …, k-1*

$$e^* = \arg\max_{e \in \mathcal{V} \setminus S_i} F(S_i \cup \{e\})$$

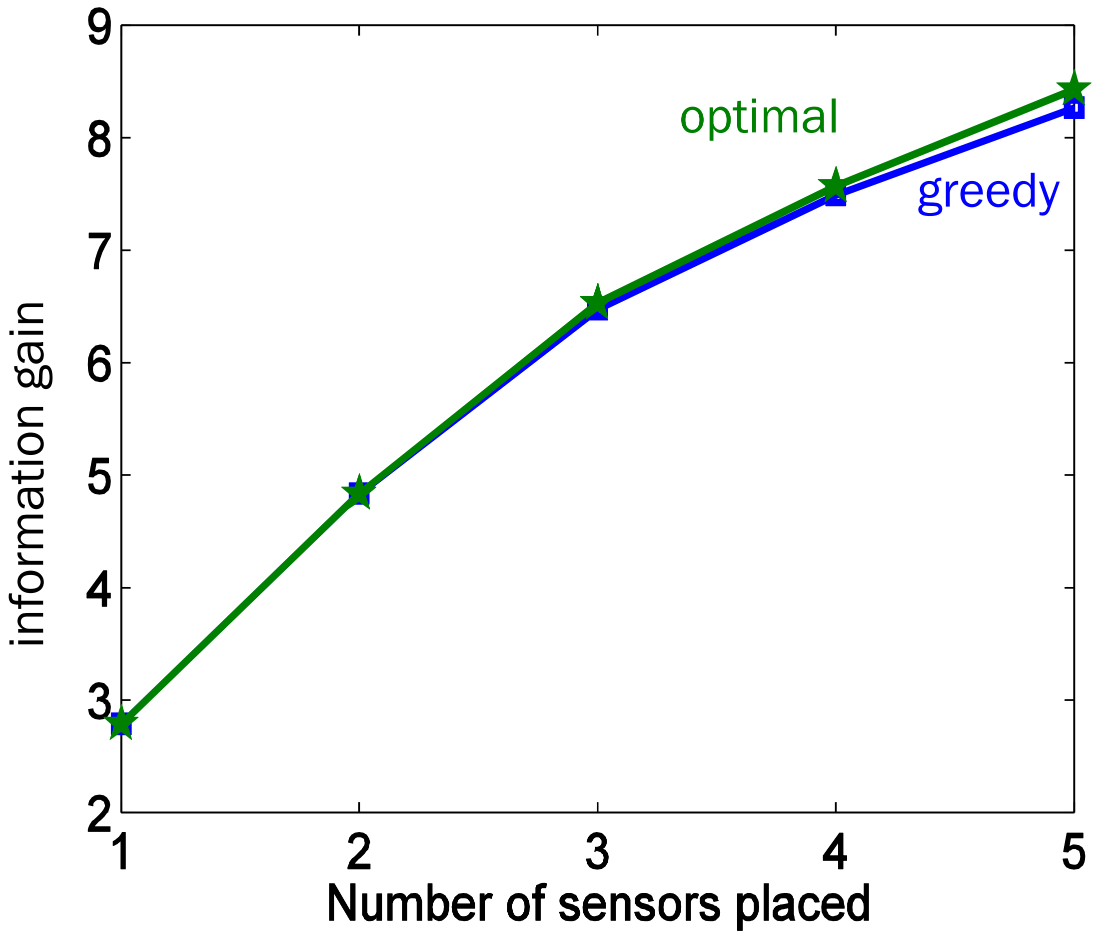
$$S_{i+1} = S_i \cup \{e^*\}$$



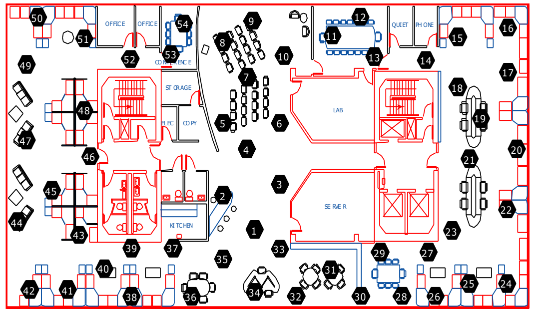
How "good" is S_k ?

How good is greedy? in practice…

empirically:



sensor placement



How good is greedy? … in theory

$$\max_S \quad F(S) \;\; \text{s.t.} \;\; |S| \leq k$$

Theorem *(Nemhauser, Wolsey, Fisher '78)*

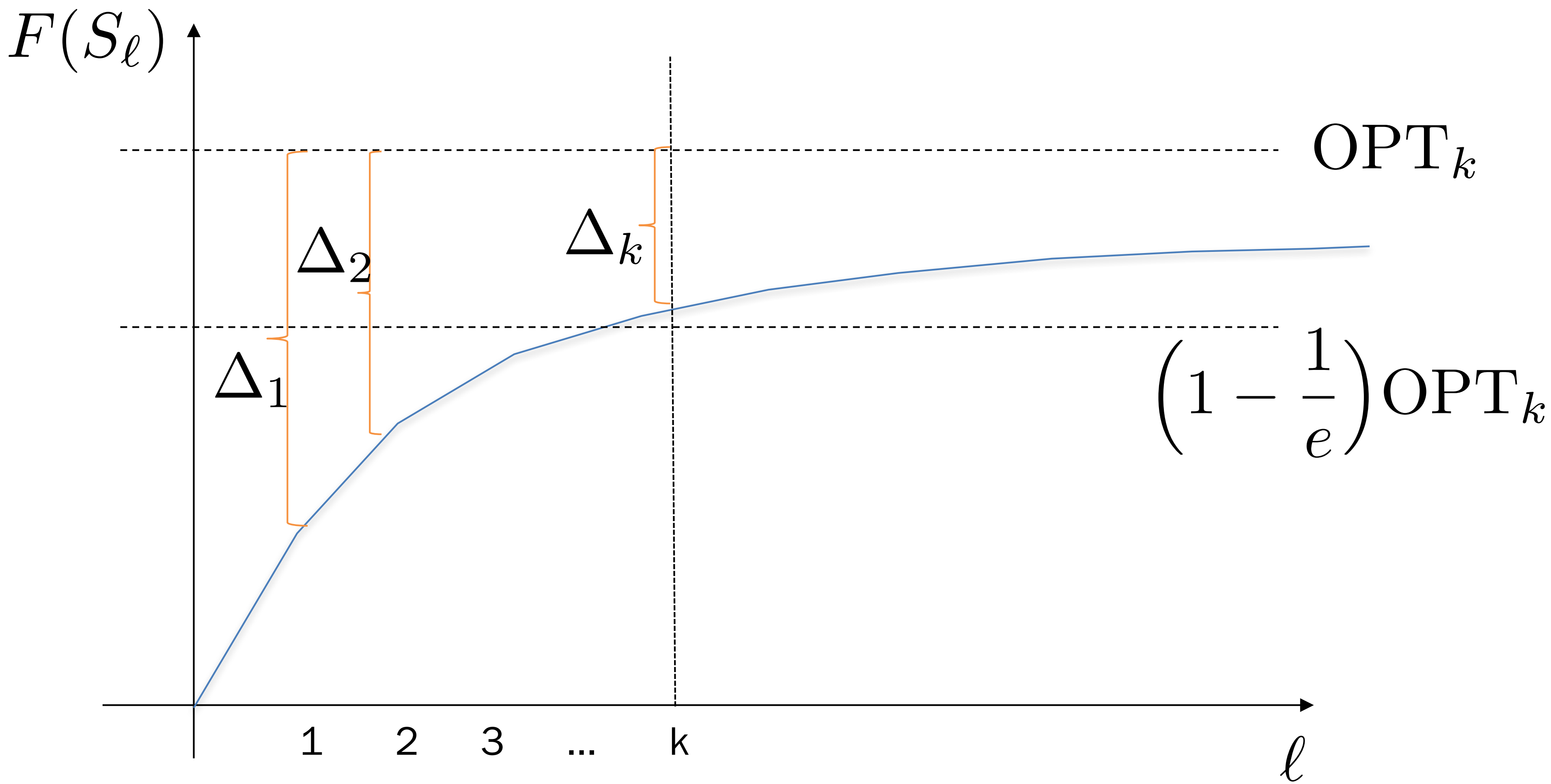
F monotone submodular, S_k solution of greedy. Then

$$F(S_k) \quad \geq \quad \left(1 - \frac{1}{e}\right) F(S^*)$$

optimal solution

in general, no poly-time algorithm can do better than that!

Proof Sketch

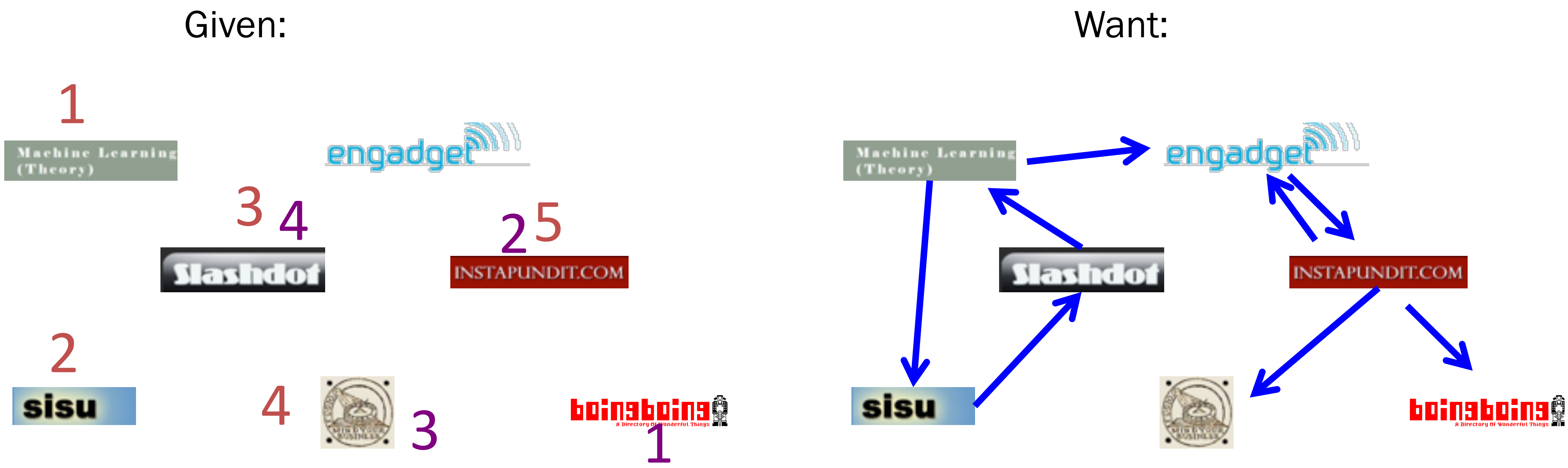


Key lemma (“Rate equation”)

$$\max_e F(S_i \cup \{e\}) - F(S_i) \geq \frac{1}{k}\Delta_i \Rightarrow \Delta_{i+1} \leq \left(1-\frac{1}{k}\right)\Delta_i \Rightarrow F(S_\ell) \geq \left(1-\left(1-\frac{1}{k}\right)^{\ell}\right)\mathrm{OPT}_k$$

Appliction: Network Inference

[Gomez Rodriguez, Leskovec, Krause ACM TKDE 2012]



Given traces of influence, wish to infer sparse directed network G=(V,E)

➔ Formulate as optimization problem

$$E^* = \arg\max_{|E| \leq k} F(E)$$

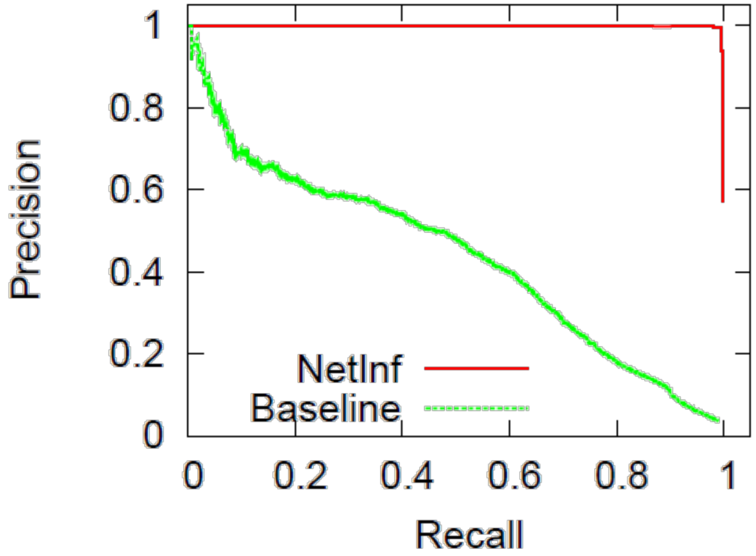
Estimation problem

- Many influence trees T consistent with data
- For cascade C_i, model $P(C_i \mid T)$
- Find sparse graph that maximizes likelihood for all observed cascades

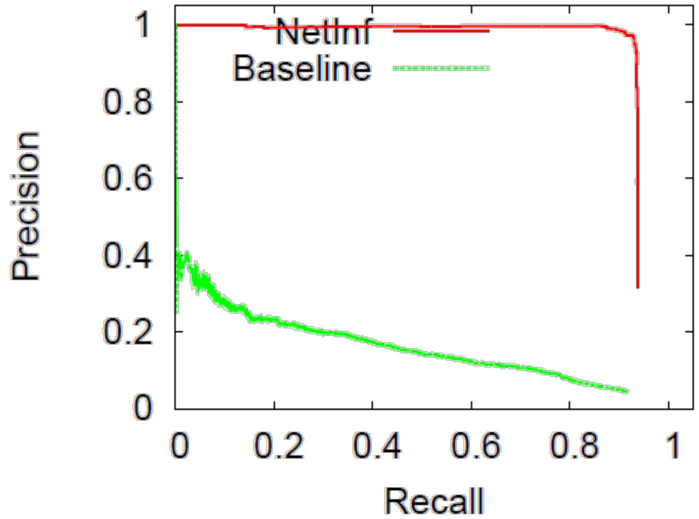
➔ Log likelihood monotonic submodular in selected edges

$$F(E) = \sum_i \log \max_{\text{tree } T \subseteq E} P(C_i \mid T)$$

Evaluation: Synthetic networks



1024 node hierarchical Kronecker exponential transmission model



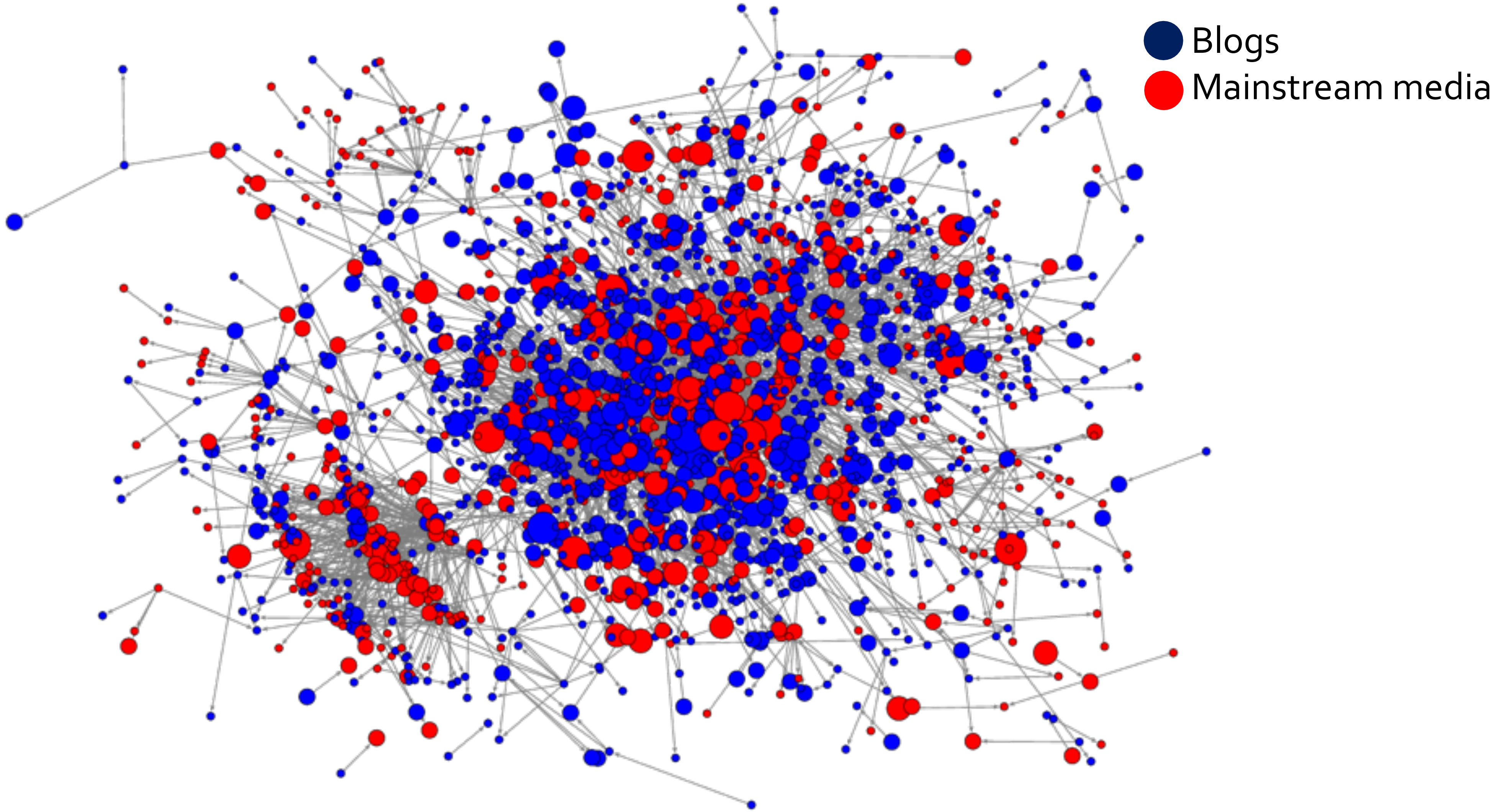
1000 node Forest Fire ($\alpha = 1.1$) power law transmission model

- Performance does not depend on the network structure:
  - Synthetic Networks: Forest Fire, Kronecker, etc.
  - Transmission time distribution: Exponential, Power Law
- Break-even point of > 90%

ETH

Diffusion Network

[Gomez Rodriguez, Leskovec, Krause ACM TKDE 2012]



Actual network inferred from 172 million articles from 1 million news sources

Questions

- What if I have more complex constraints?
- Greedy takes $O(nk)$ time. What if n, k are large?
- What if my function is not monotone?

More complex constraints: budget

$$\max \quad F(S) \quad \text{s.t.} \quad \sum_{e \in S} c(e) \leq B$$

1. run greedy: S_{gr}
2. run a modified greedy: S_{mod}

$$e^* = \arg\max_e \frac{F(S_i \cup \{e\}) - F(S_i)}{c(e)}$$

3. pick better of S_{gr}, S_{mod}

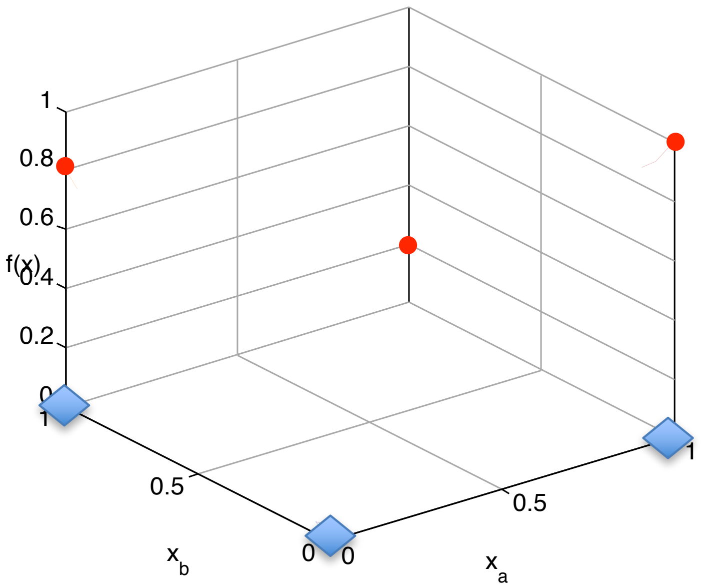
→ approximation factor: $1 - \frac{1}{\sqrt{e}}$

even better but less fast: partial enumeration *(Sviridenko, '04)* or filtering *(Badanidiyuru & Vondrák '14)*

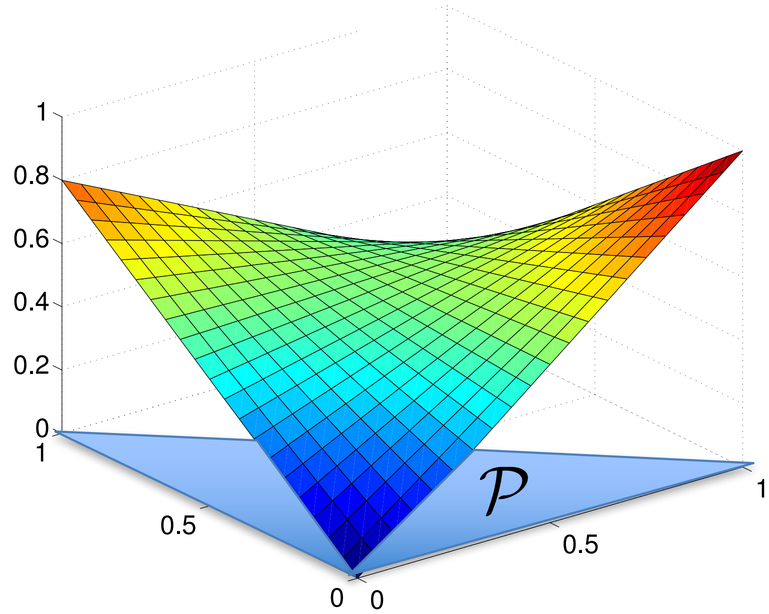
(Leskovec-Krause-Guestrin-Faloutsos-VanBriesen-Glance '07)

Relax: Discrete to continuous

$\max F(S)$



$\max \quad f_M(x)$



Algorithm:

1. approximately maximize f_M over $\mathcal{P} = \mathrm{conv}(\mathcal{I})$
2. round to discrete set

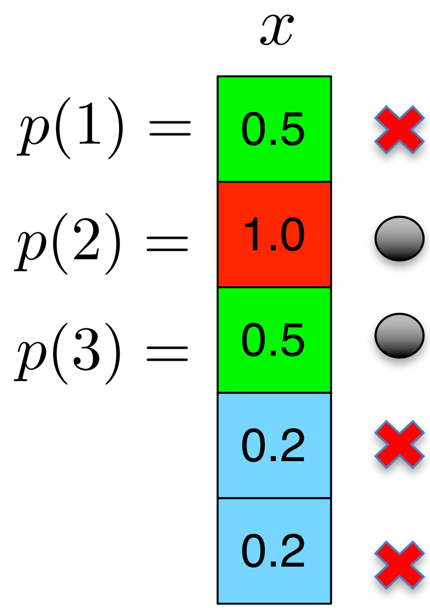
(Vondrák '08; Calinescu-Chekuri-Pal-Vondrák '11; Kulik-Shachnai-Tamir'11)

Multilinear extension

sample item e with probability x_e

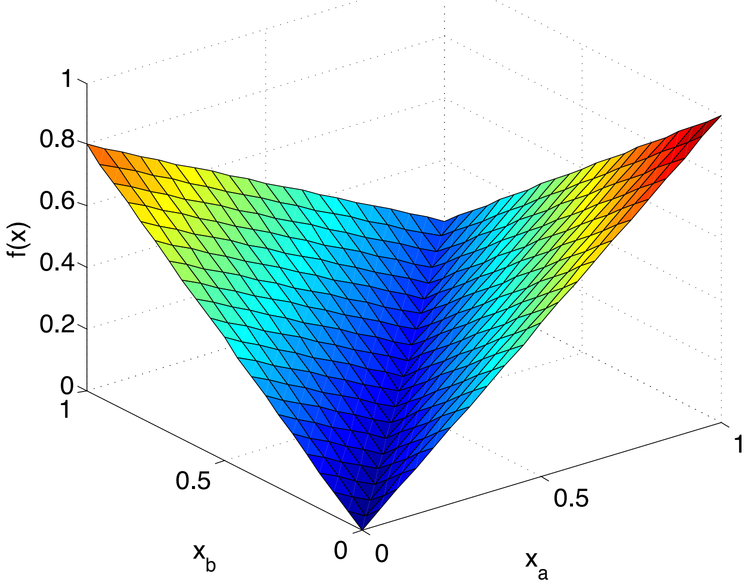
$$f_M(x) = \mathbb{E}_{S\sim x}\left[F(S)\right]$$

$$= \sum_{S\subseteq\mathcal{V}} F(S) \prod_{e\in S} x_e \prod_{e\notin S}(1 - x_e)$$



Compare: Multilinear vs. Lovász ext.

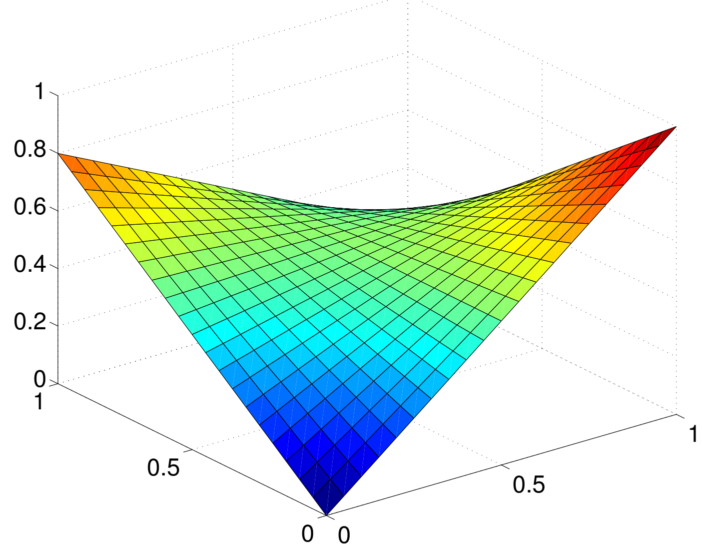
$$f_L(x) = \mathbb{E}_{S\sim\theta}\left[F(S)\right]$$



- convex
- computable in O(n log n)

➢ Submodular minimization

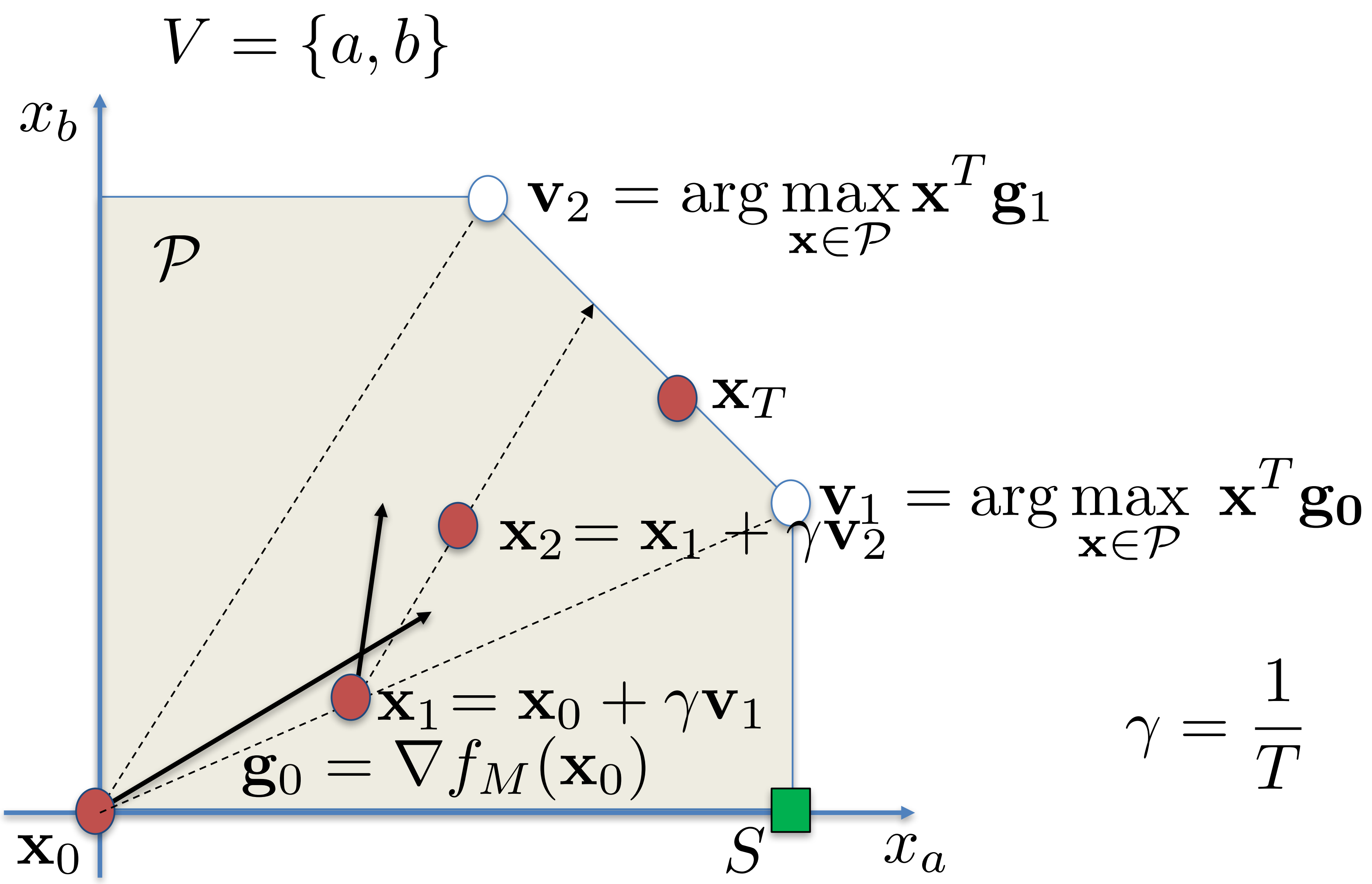
$$f_M(x) = \mathbb{E}_{S\sim x}\left[F(S)\right]$$



- concave in certain directions, convex in others
- approximate by sampling

➢ Submodular maximization

Illustration of Continuous Greedy



Continuous submodular maximization

- Continuous Greedy (~Frank Wolfe) “works” for any
  - downward closed solvable polytope *P* *(Calinescu-Chekuri-Pál-Vondrák’11)*
  - monotone continuous “DR-submodular” function (beyond multilinear extension) *(Bian-Mirzasoleiman-Buhmann-Krause’16)*

    ➔ Non-convex optimization with guarantees
- ”works” means (1-1/e) approx. for continuous problem
- Rounding strategy depends on constraints
  - Pipage rounding for matroids *(Ageev, Sviridenko ’04)*
  - Contention resolution for more general *P* *(Chekuri-Vondrák-Zenklusen’11)*

Questions

- What if I have more complex constraints?
  - budget constraints
  - Downward closed constraints (matroids, p-systems, knapsacks, their intersections, …)
- Greedy takes *O(nk)* time. What if *n*, *k* are large?
- What if my function is not monotone?

Scaling up the greedy algorithm [Minoux' 78]

In round i+1,

- have picked $A_i = \{s_1,\dots,s_i\}$
- pick $s_{i+1} = \text{argmax}_s\, F(A_i \cup \{s\})-F(A_i)$

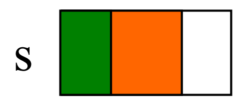
I.e., maximize "marginal benefit" $\Delta(s \mid A_i)$

$$\Delta(s \mid A_i) = F(A_i \cup \{s\})-F(A_i)$$

Key observation: Submodularity implies

$$i \leq j \;\Rightarrow\; \Delta(s \mid A_i) \geq \Delta(s \mid A_j)$$

$\Delta(s \mid A_i) \geq \Delta(s \mid A_{i+1})$

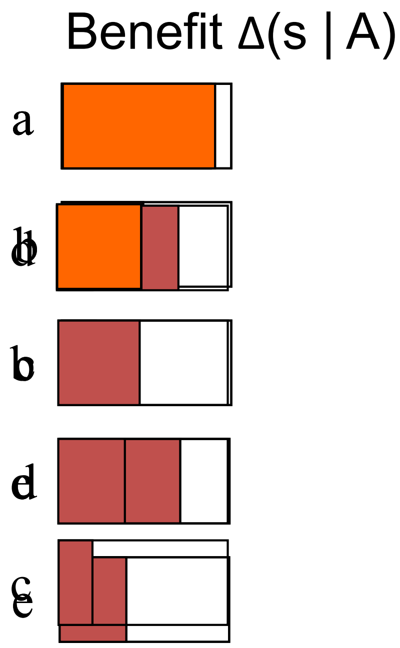


Marginal benefits can never increase!

"Lazy" greedy algorithm [Minoux '78]

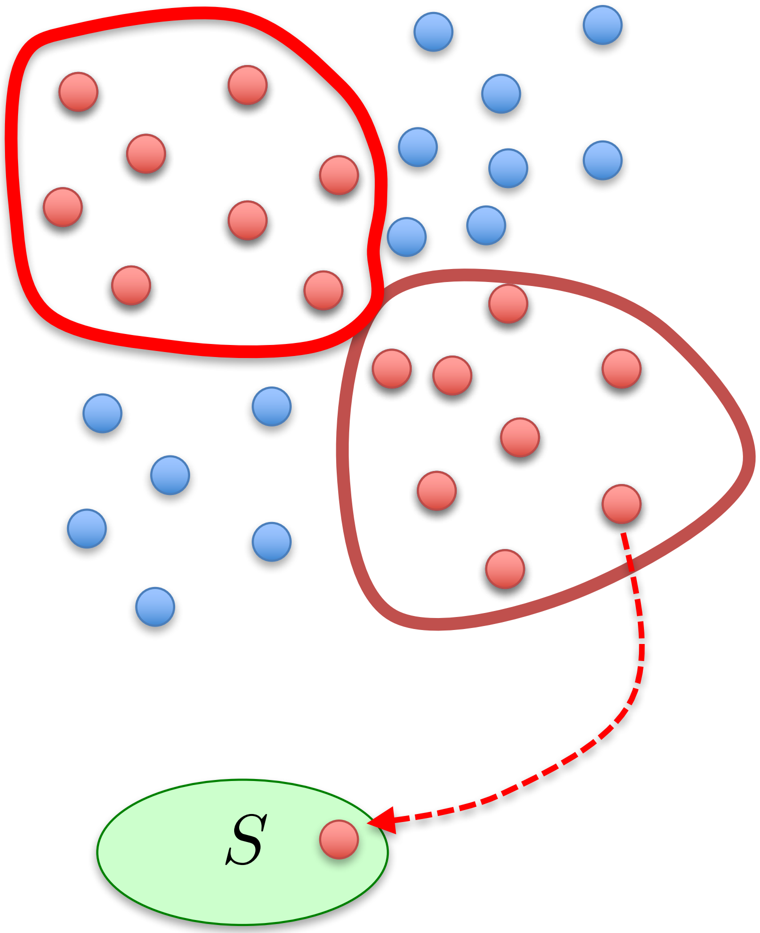
Lazy greedy algorithm:

- First iteration as usual
- Keep an ordered list of marginal benefits Δ_i from previous iteration
- Re-evaluate Δ_i only for top element
- If Δ_i stays on top, use it, otherwise re-sort



Note: Very easy to compute online bounds, lazy evaluations, etc. [Leskovec, Krause et al. '07]

Lazier than lazy greedy



(Mirzasoleiman et al 2014)

$$\max_S \; F(S) \text{ s.t. } |S| \leq k$$

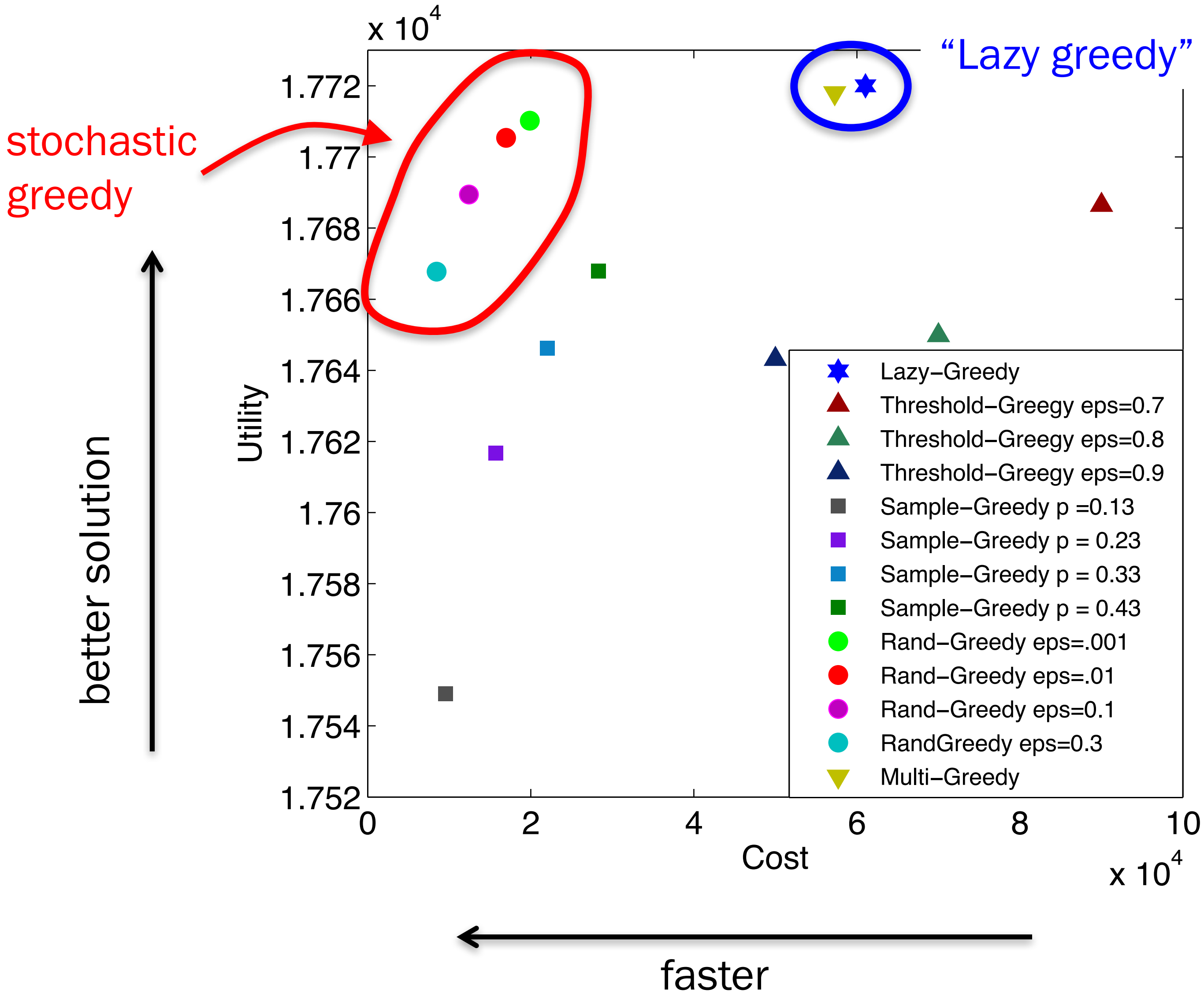
for i=1…k:

- randomly pick set *T* of size $\frac{n}{k} \log \frac{1}{\epsilon}$
- find best *a* element in *T* and add

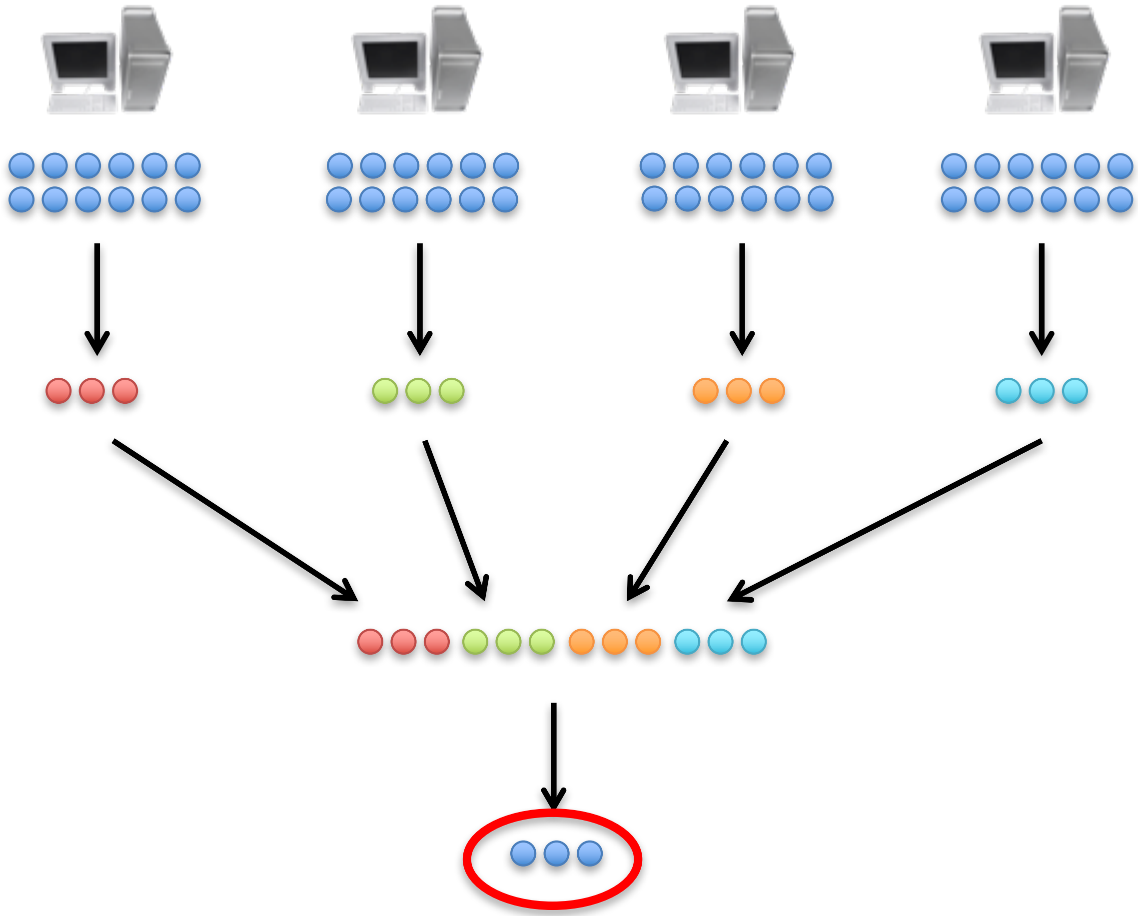
$$a_i = \arg\max_{a \in T} F(a|S_{i-1})$$

$$S_i \leftarrow S_{i-1} \cup \{a_i\}$$

Performance



Distributed greedy algorithms



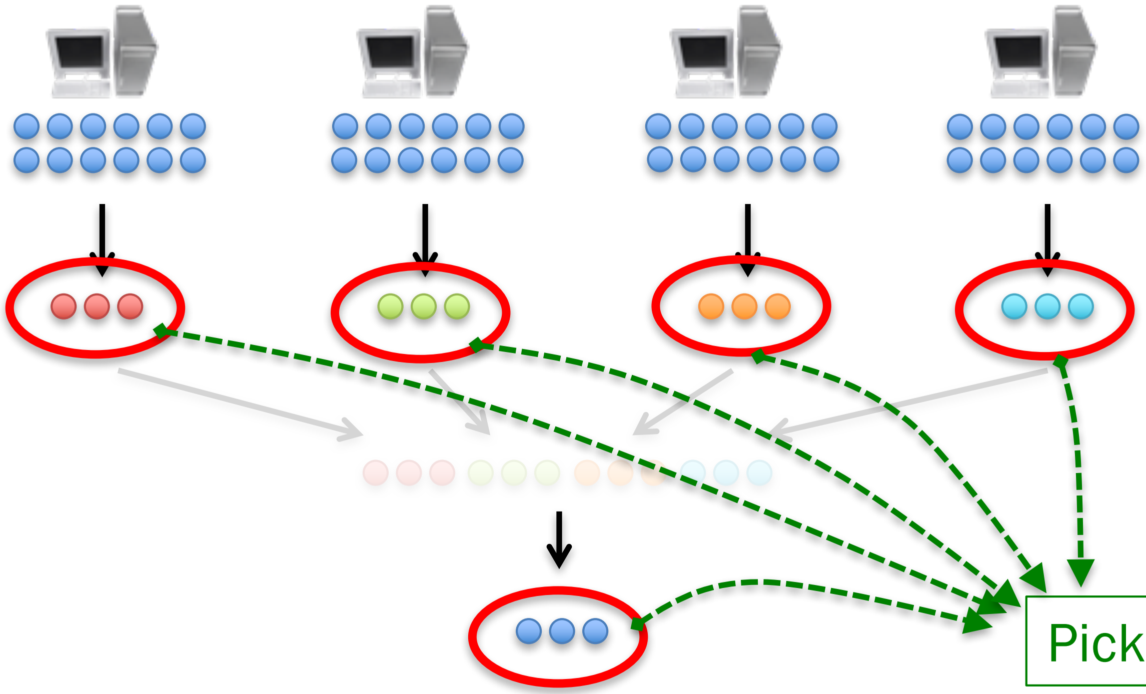
greedy is sequential.
pick in parallel??

pick *k* elements
on each machine.

combine and run
greedy again.

Is this useful?

GreeDI



pick in parallel
from *m* machines

Is this useful?

Pick the best of m+1 solutions

Approximation factor:

$$\frac{1}{\min\{\sqrt{k}, m\}}$$

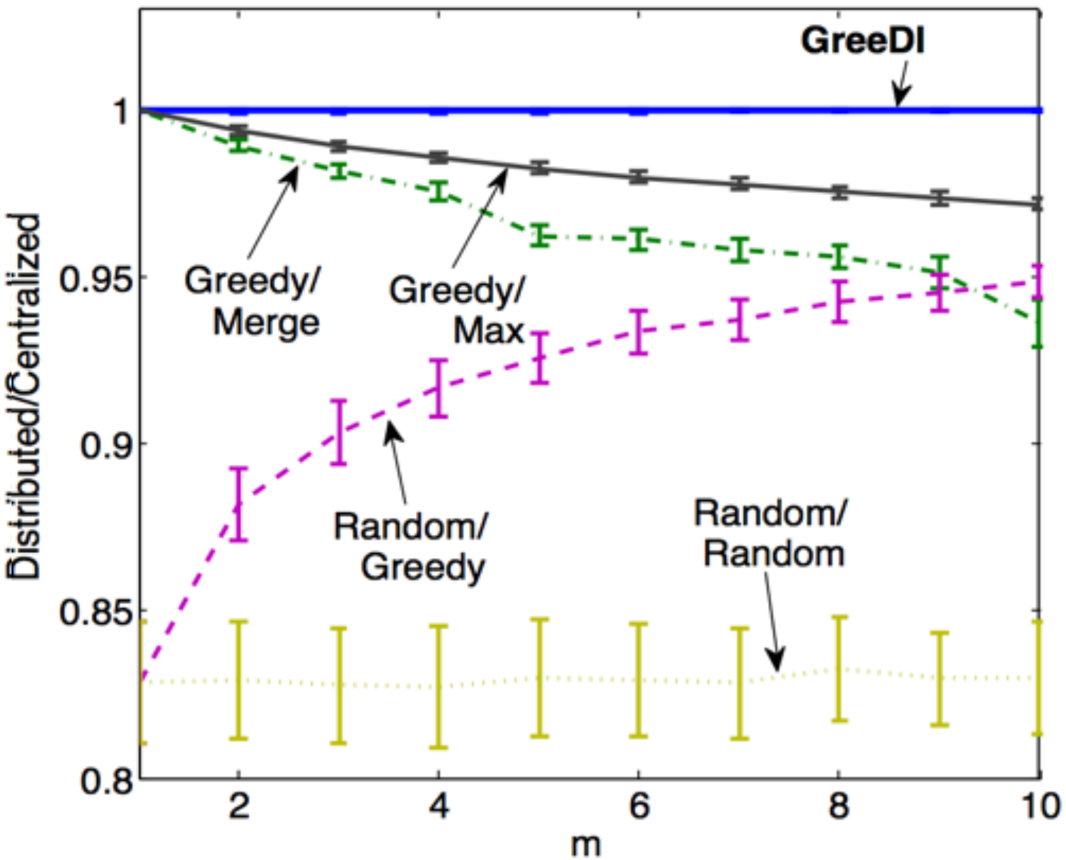
better with geometric structure

New approximation factor:

$$\frac{1}{2}\left(1 - \frac{1}{e}\right)$$

(Mirzasoleiman-Karbasi-Sarkar-Krause’13, da Ponte Barbosa-Ene-Nguyen-Ward’15)

Empirical Performance



(Mirzasoleiman-Karbasi-Sarkar-Krause '13)

Questions

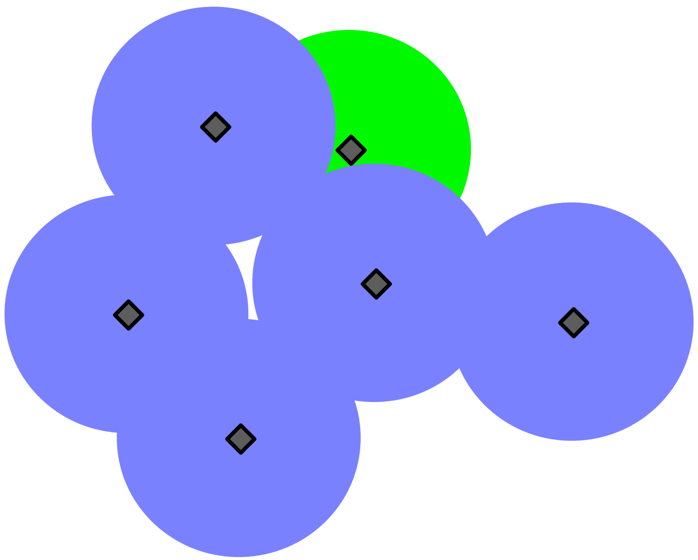
- What if I have more complex constraints?
  - budget constraints
  - Downward closed solvable polytopes

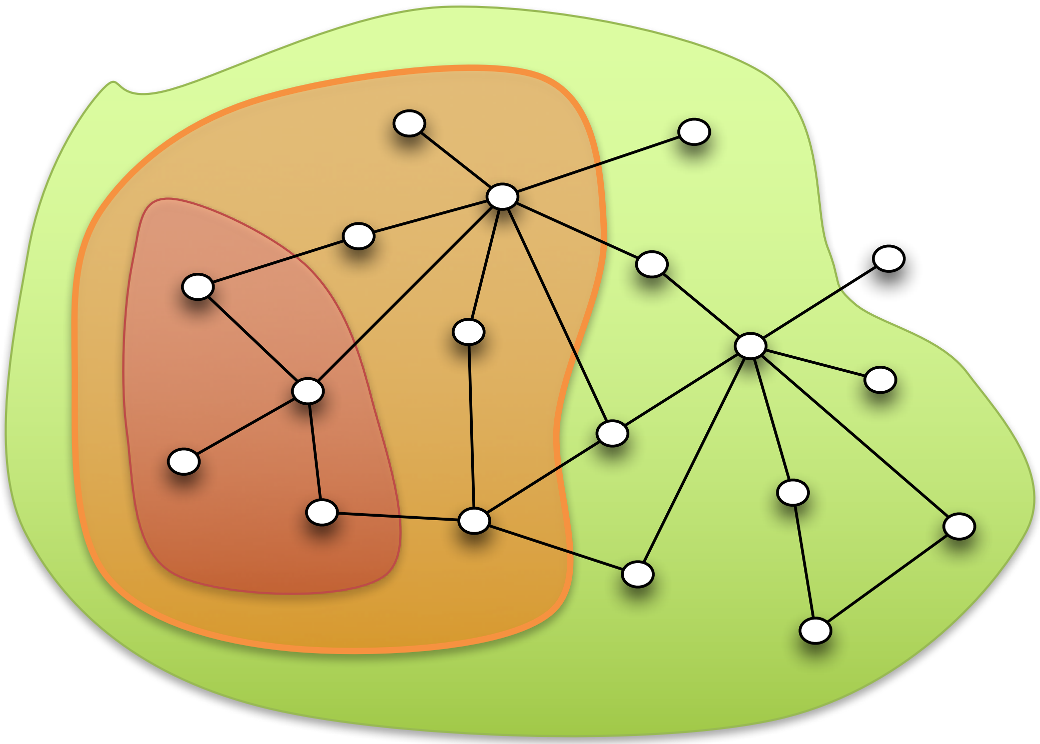
- Greedy takes *O(nk)* time. What if n, k are large?
  - Lazy greedy, lazier than lazy greedy *(Minoux'78, Mirzasoleiman-Badanidiyuru-Karbasi-Vondrák-Krause'15)*
  - filtering / streaming / multi-stage *(Badanidiyuru & Vondrák 2014; Badanidiyuru-Mirzasoleiman-Karbasi-Krause'14,* Wei-Iyer-Bilmes'14)
  - Distributed *(Mirzasoleiman-Karbasi-Sarkar-Krause'13, Kumar-Moseley-Vassilivitskii-Vattani'13)*

- What if my function is not monotone?

Non-monotone functions

$$\cancel{\text{if } \; S \subseteq T \quad \text{then} \quad F(S) \leq F(T)}$$





3 5 1

Greedy can fail ...

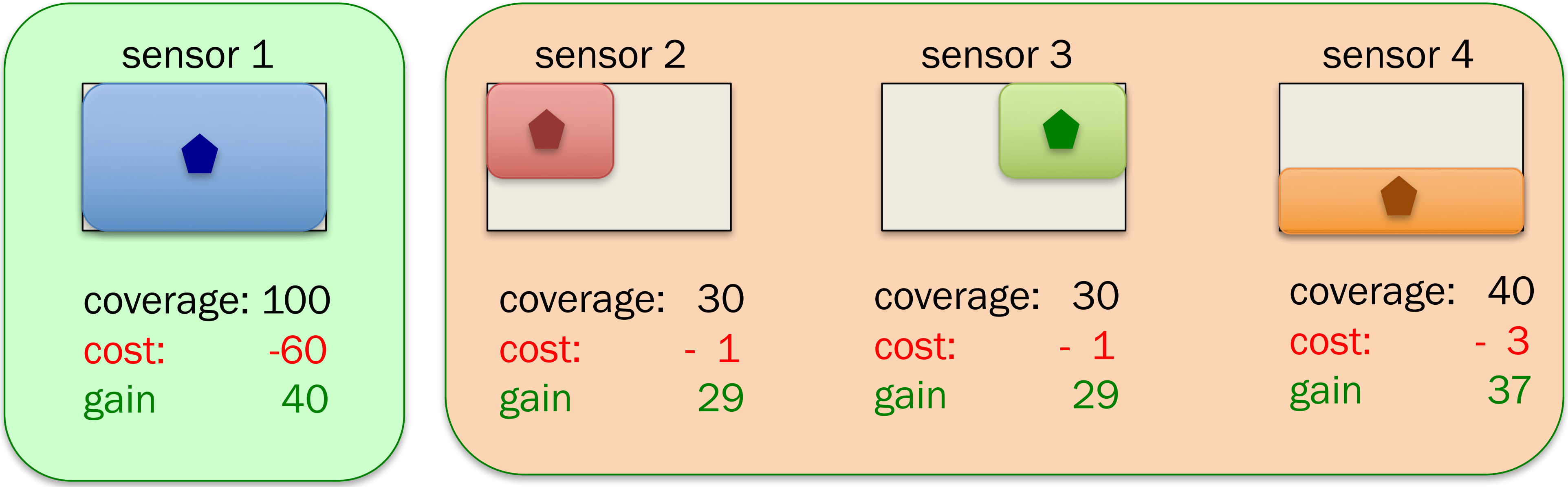
$$F(A) = \left| \bigcup_{a \in A} \text{area}(a) \right| - \sum_{a \in A} c(a)$$

greedy

$F(A$

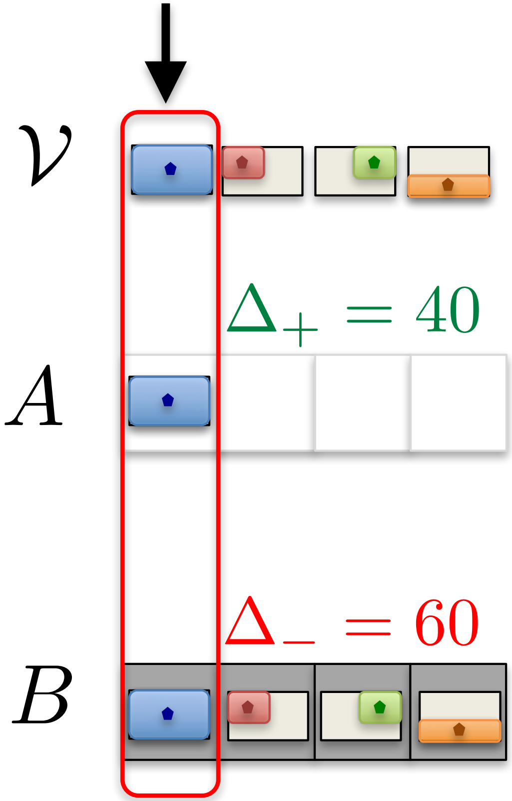
optimal solution

$$F(A) = 95$$



$$S_0 = \emptyset \qquad S_1 = \emptyset \cup \arg\max_{a \in \mathcal{V}} F(a)$$

Double (bidirectional) greedy



Start: $A = \emptyset,\; B = \mathcal{V}$

for *i=1, ..., n* *//add or remove?*

- gain of adding (to A):

$$\Delta_+ = [\,F(A \cup a_i) - F(A)\,]_+$$

- gain of removing (from B):

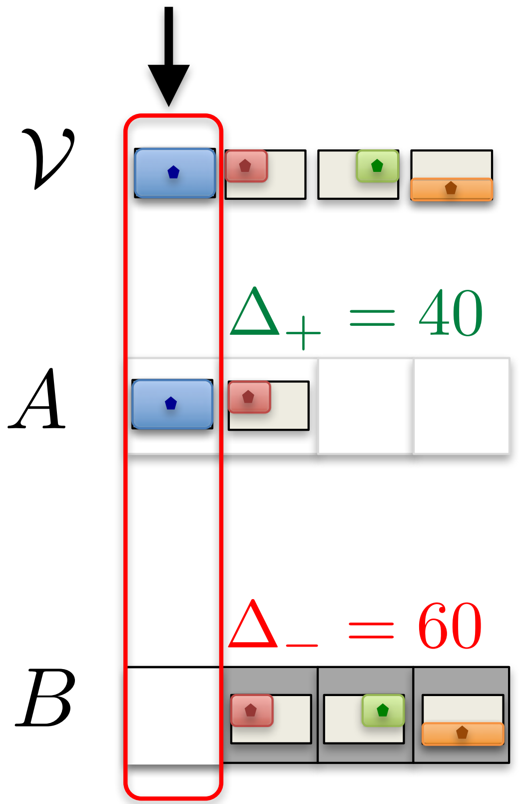
$$\Delta_- = [\,F(B \setminus a) - F(B)\,]_+$$

add with probability

$$\mathbb{P}(\text{add}) = \frac{\Delta_+}{\Delta_+ + \Delta_-}$$ = 40%

coverage: 100
cost: -60

Double (bidirectional) greedy



Start: $A = \emptyset, \; B = \mathcal{V}$

for *i=1, ..., n* *//add or remove?*

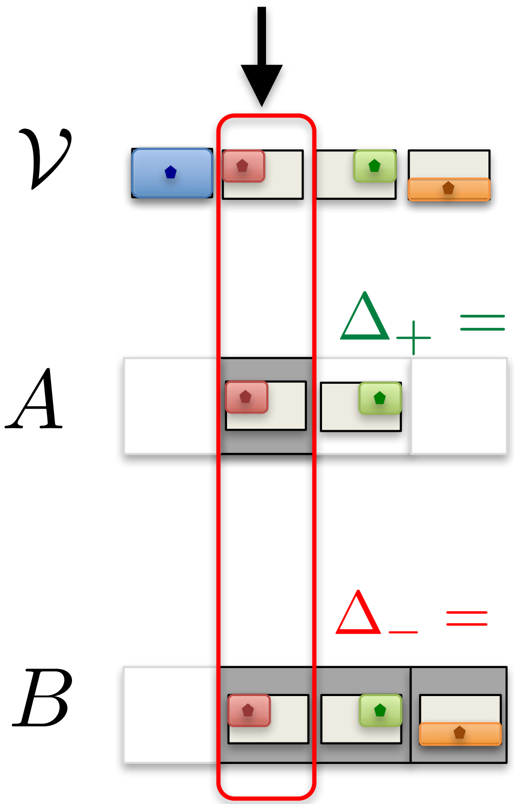
add with probability

$$\mathbb{P}(\text{add}) = \frac{\Delta_+}{\Delta_+ + \Delta_-}$$

add to A or $\boxed{\text{remove from B}}$

coverage: 100
cost: -60

Double (bidirectional) greedy



$\Delta_+ = 29$

$\Delta_- = [-29]_+ = 0$

Start: $A = \emptyset,\ B = \mathcal{V}$

for *i=1, ..., n* *//add or remove?*

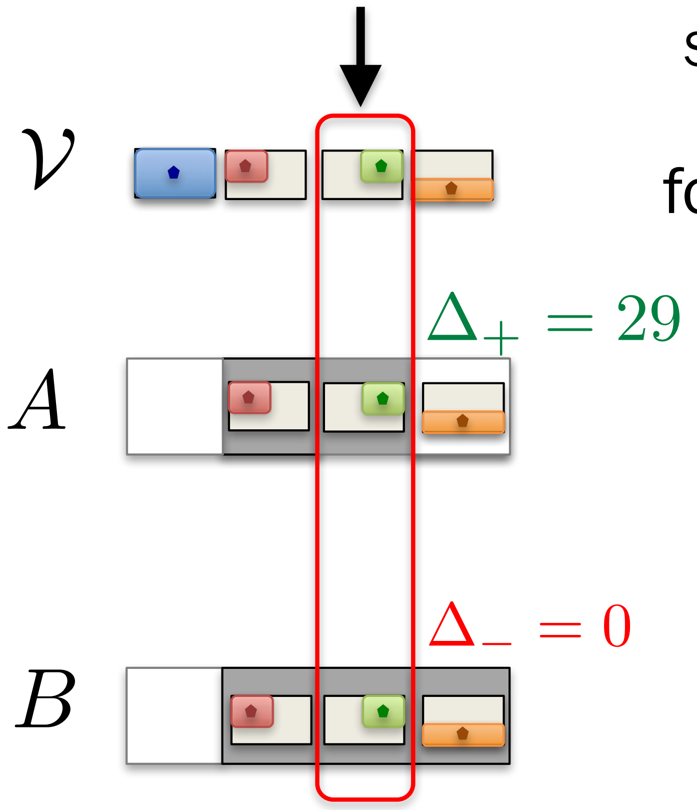
add with probability

$$\mathbb{P}(\text{add}) = \frac{\Delta_+}{\Delta_+ + \Delta_-} = \frac{29}{29}$$

$\boxed{\text{add to A}}$ or remove from B

coverage:	30
cost:	- 1

Double (bidirectional) greedy



Start: $A = \emptyset, \ B = \mathcal{V}$

for *i=1, ..., n* *//add or remove?*

add with probability

$$\mathbb{P}(\text{add}) = \frac{\Delta_+}{\Delta_+ + \Delta_-} \quad = \frac{39}{40}$$

add to A or remove from B

coverage:	30
cost:	- 1

Double greedy

$$\max_{S \subseteq \mathcal{V}} \; F(S)$$

Theorem *(Buchbinder, Feldman, Naor, Schwartz '12)*

F submodular, S_g solution of double greedy. Then

$$\mathbb{E}[F(S_g)] \geq \frac{1}{2} F(S^*)$$

optimal solution

ETH

Non-monotone maximization

- Generally inapproximable unless F is nonnegative
- Unconstrained maximization:
  - Local search *(Feige-Mirrokni-Vondrák'07)*
  - Double greedy: Optimal ½ approximation
    (Buchbinder-Feldman-Naor-Schwartz'12)
- Constrained maximization:
  - Cardinality constraints: randomized greedy *(Buchbinder-Feldman-Naor-Schwartz'14)*
  - Filtering based algorithms *(Mirzasoleiman-Badanidiyuru-Karbasi'16)*
  - More general constraints: Continuous local search via multilinear extension *(Chekuri—Vondrák-Zenklusen'11)*
- Distributed algorithms? yes!
  - divide-and-conquer as before *(de Ponte Barbosa-Ene-Nguyen-Ward '15)*
  - concurrency control / Hogwild *(Pan-Jegelka-Gonzalez-Bradley-Jordan '14)*

Submodular maximization: summary

- Many applications: diverse, informative subsets
- NP-hard, but variants of greedy / local search work
- Distinguish monotone / non-monotone
- Can handle several types of constraints
- Scalable algorithms for solving massive problems

Summary: Submodular Optimization

Minimization	Maximization
Unconstrained SFMin tractable, constrained SFMin generally hard	SFMax generally hard distinguish montone & non-monotone
Combinatorial and continuous algorithms	Greedy-like and continuous algorithms
Convex Lovász extension	Nonconvex multilinear extension
Faster algorithms for special cases	Fast distributed/ streaming algorithms