Massachusetts
Institute of
Technology

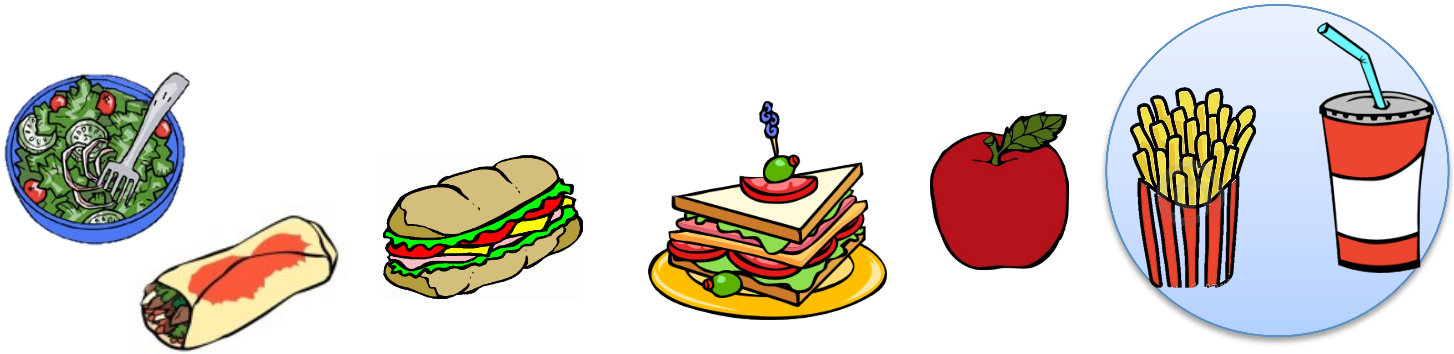
Submodular Functions – Part I

ML Summer School Cádiz

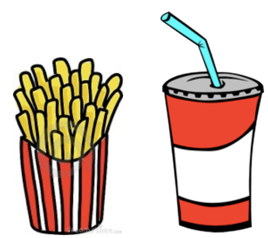
Stefanie Jegelka
MIT

Set functions

ground set

$\mathcal{V} = \{$  $\}$

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

$F($  $) =$ cost of buying items together, or

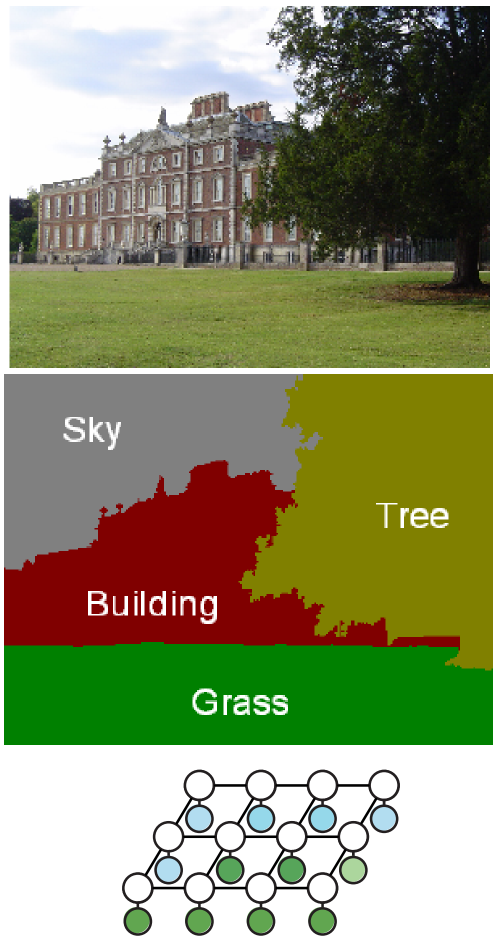
utility, or

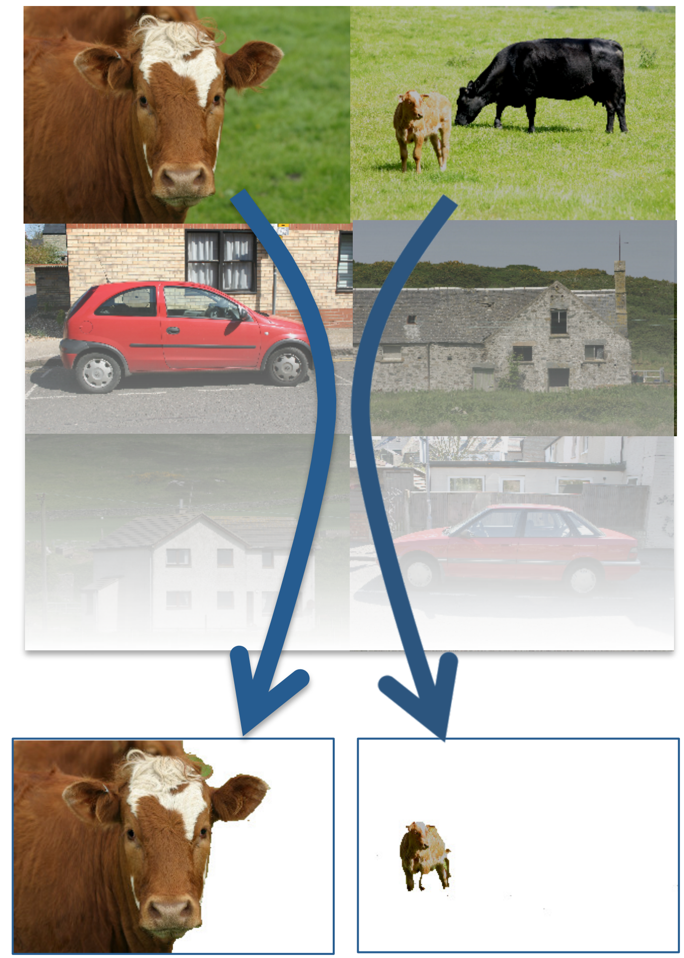
probability, ...

We will assume:

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

Discrete Labeling



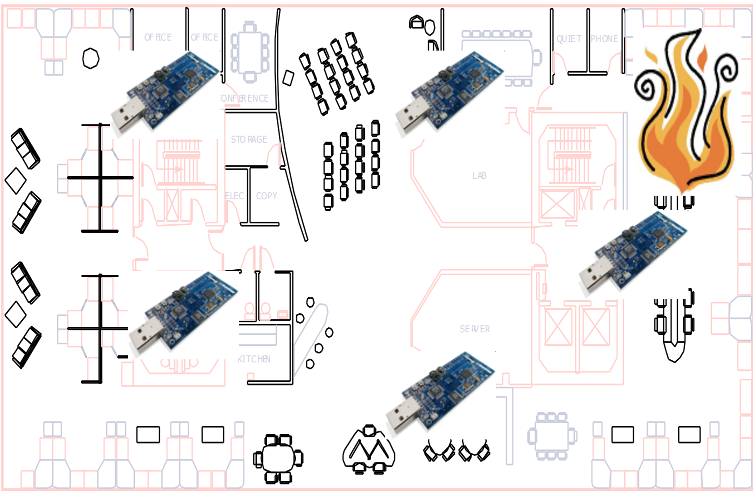


$$F(S) = \text{coherence} \ + \ \text{likelihood}$$

Summarization

$$F(S) = \text{relevance} + \text{diversity or coverage}$$

Informative Subsets

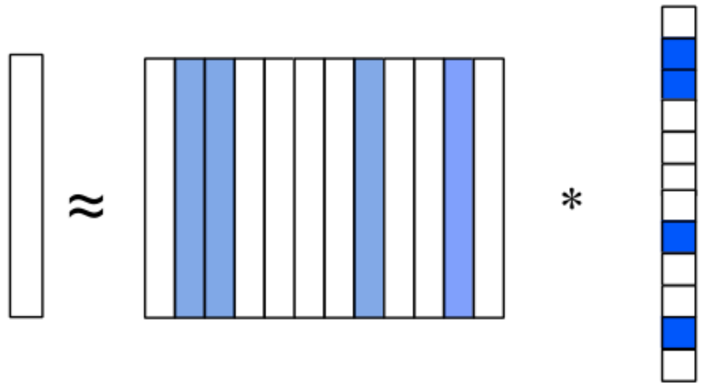


- where put sensors?
- which experiments?
- summarization

$$F(S) = \text{“information”}$$

Sparsity

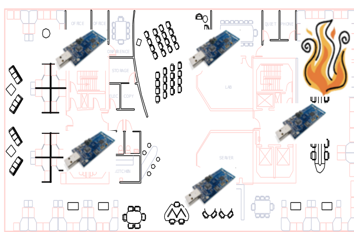
$$y = A\,x \quad + \quad \text{noise}$$

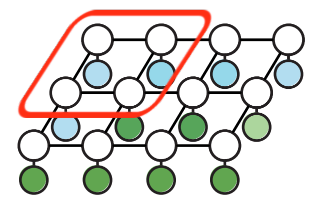


$F(S) =$ "penalty on support pattern"

Formalization

- Formalization:
  Optimize a set function F(S) (under constraints)



- generally very hard ☹
- submodularity helps:
  efficient optimization & inference with guarantees! ☺

Roadmap

- Submodular set functions
  - what is this? where does it occur? how recognize?

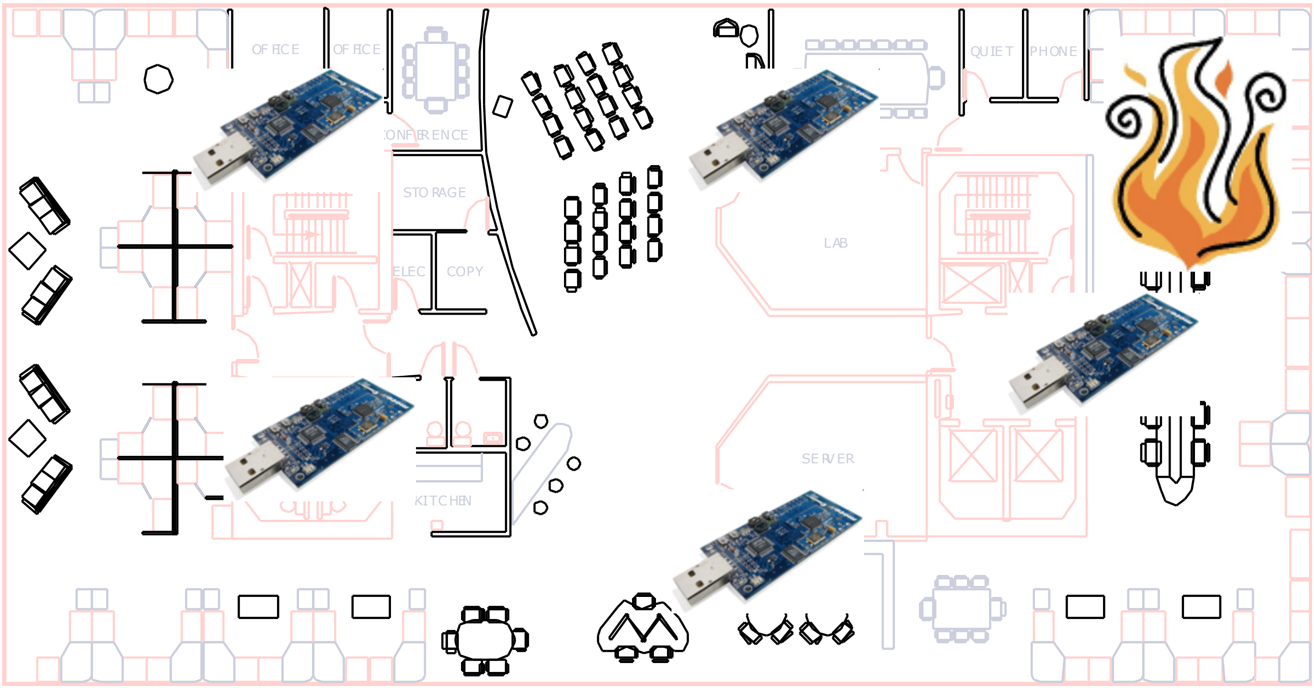
- Maximizing submodular functions:
  diversity, repulsion, concavity
  greed is not too bad

- Minimizing submodular functions:
  coherence, regularization, convexity
  the magic of "discrete analog of convex"

- Other questions around submodularity & ML

more reading & papers: http://people.csail.mit.edu/stefje/mlss/literature.pdf

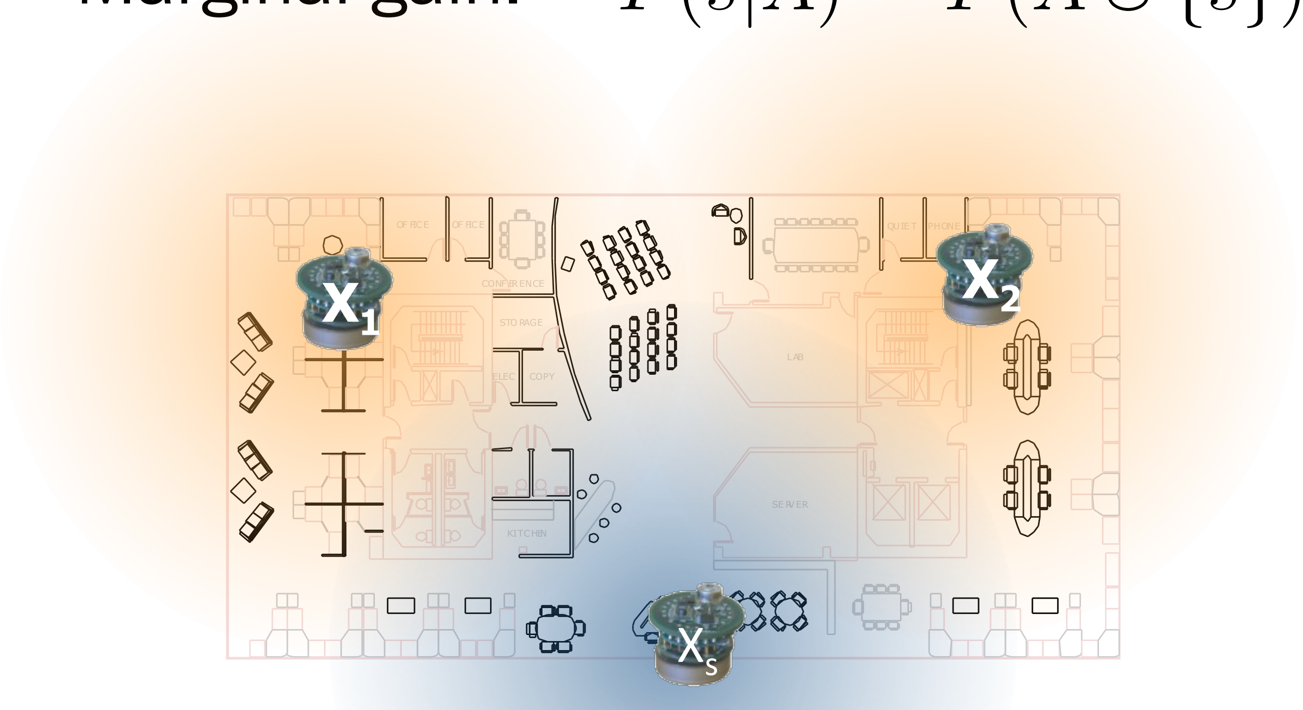
Sensing



$\mathcal{V}$ = all possible locations
F(S) = information gained from locations in *S*

Marginal gain

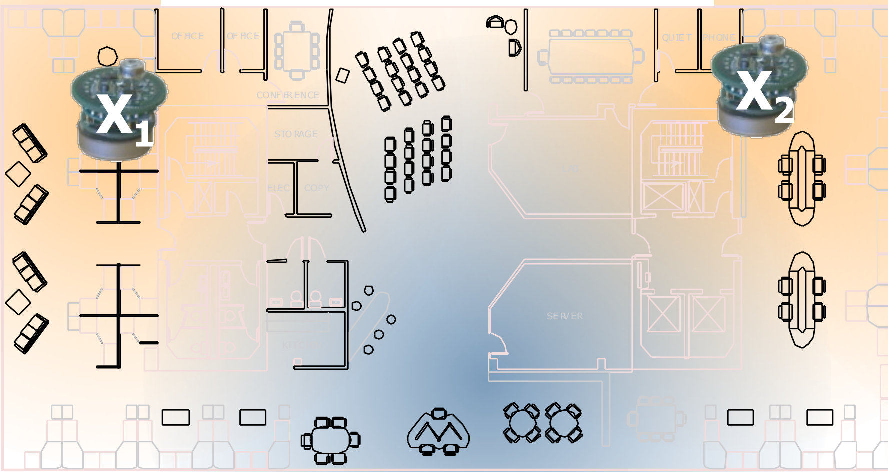
- Given set function $F : 2^V \rightarrow \mathbb{R}$
- Marginal gain: $F(s|A) = F(A \cup \{s\}) - F(A)$



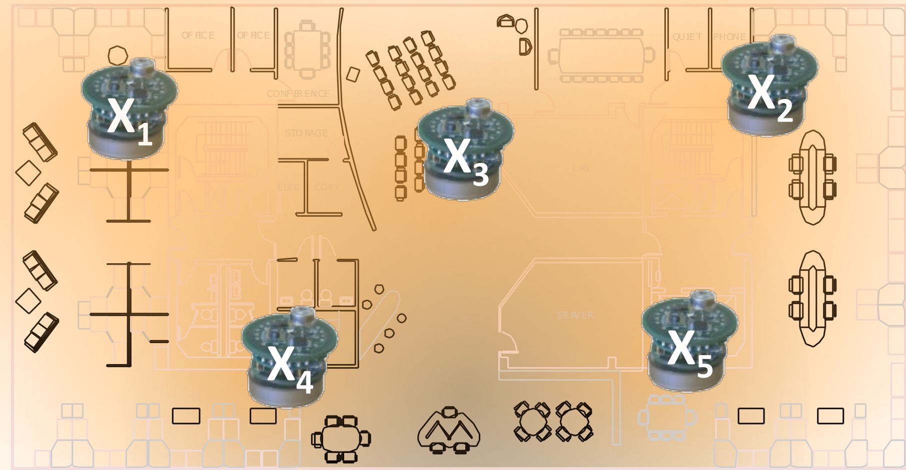
new sensor s

Diminishing marginal gains

placement A = {1,2}



placement B = {1,…,5}

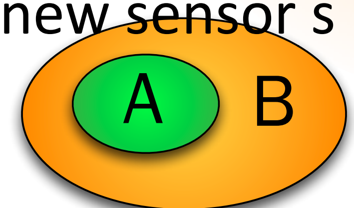


Big gain

new sensor s

small gain

+ • s



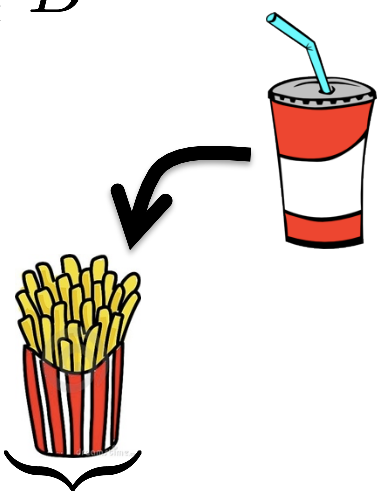
+ • s

$$A \subseteq B$$

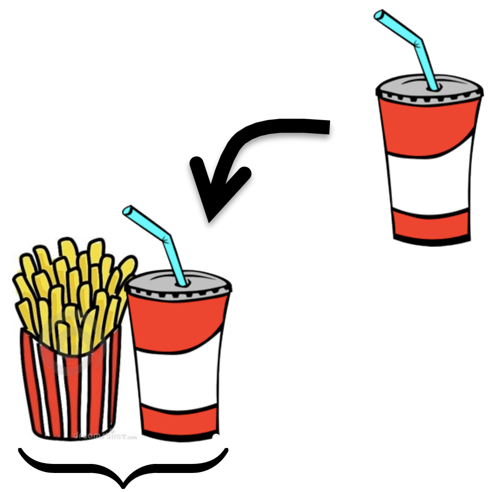
$$F(A \cup s) - F(A) \quad \geq \quad F(B \cup s) - F(B)$$

Submodularity

$A \subseteq B$



A



B

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

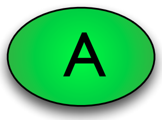
extra cost:
one drink

extra cost:
free refill ☺

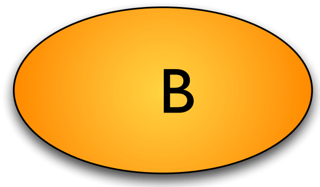
diminishing marginal costs

Submodular set functions

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



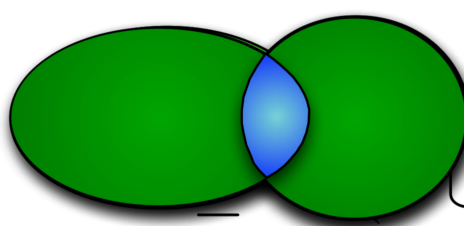
\+ • e



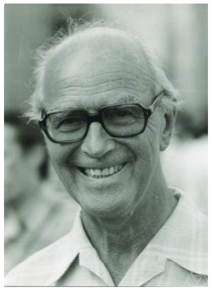
\+ • e

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

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

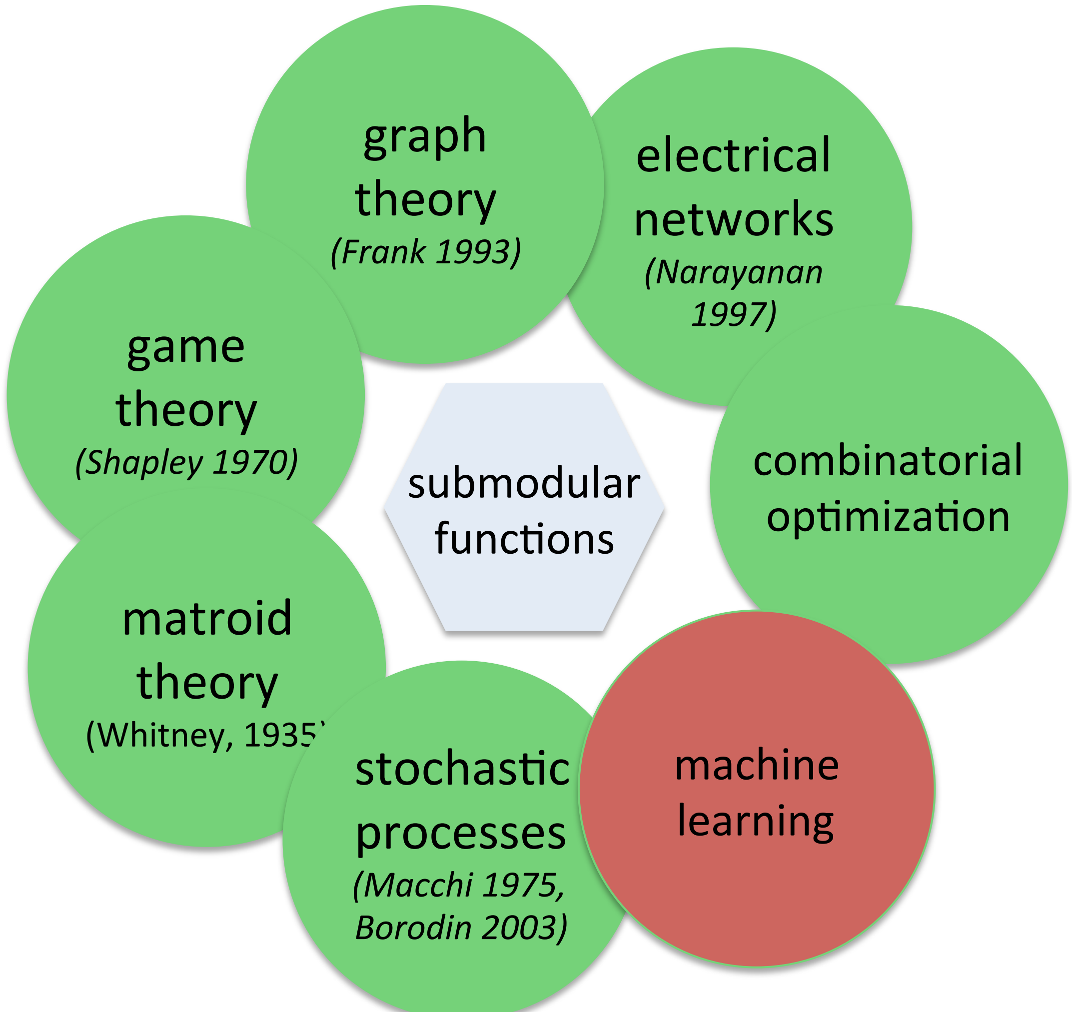
$F(S) \;+\; F(T)$  $\cup T) \;+\; F(S \cap T)$

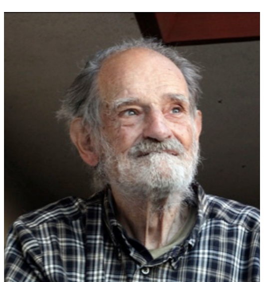
The big picture



G. Choquet

J. Edmonds



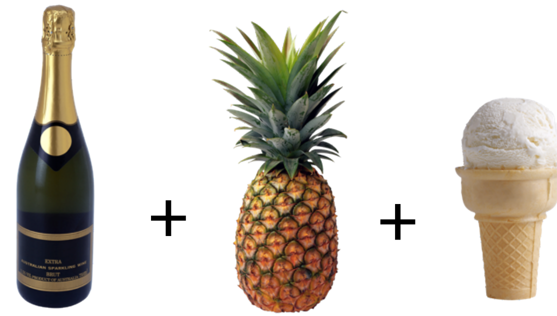


L.S. Shapley

L. Lovász

Examples

- each element e has a weight $w(e)$



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

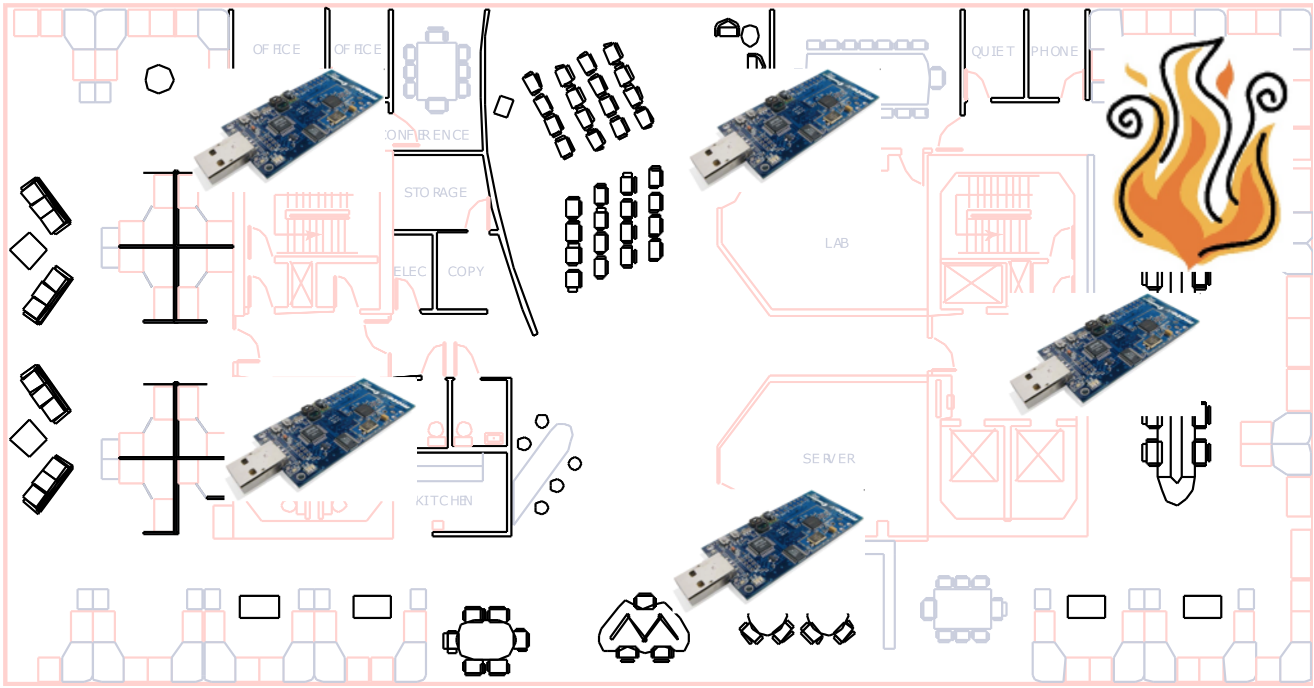
$A \subset B$

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

linear / modular function

F and –*F* always submodular!

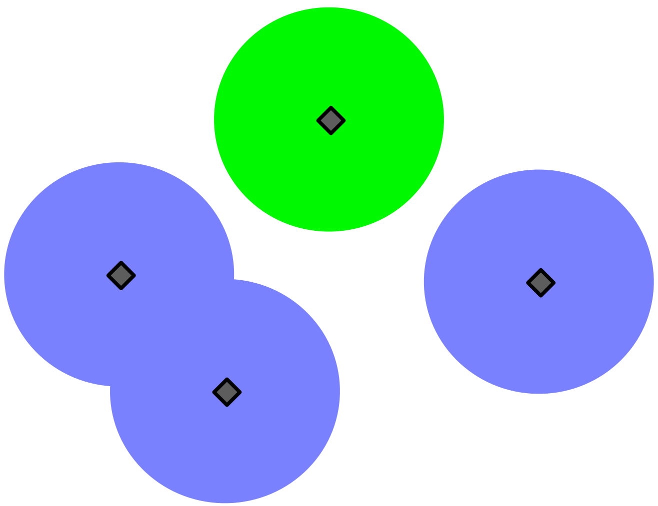
Examples

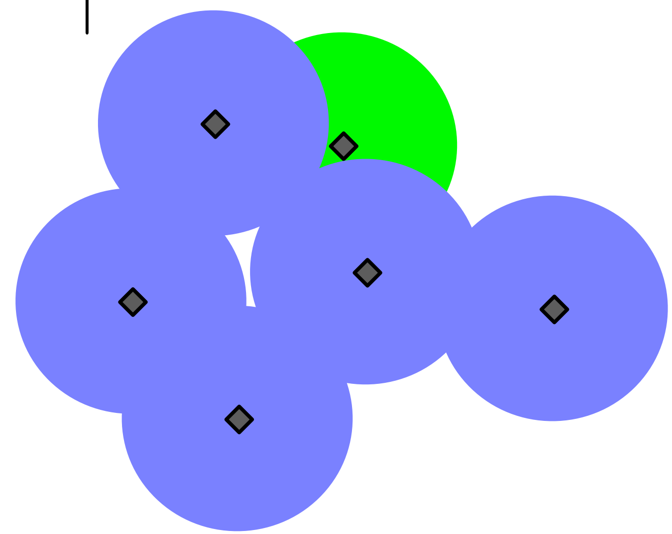


sensing:
F(S) = information gained from locations S

Example: cover

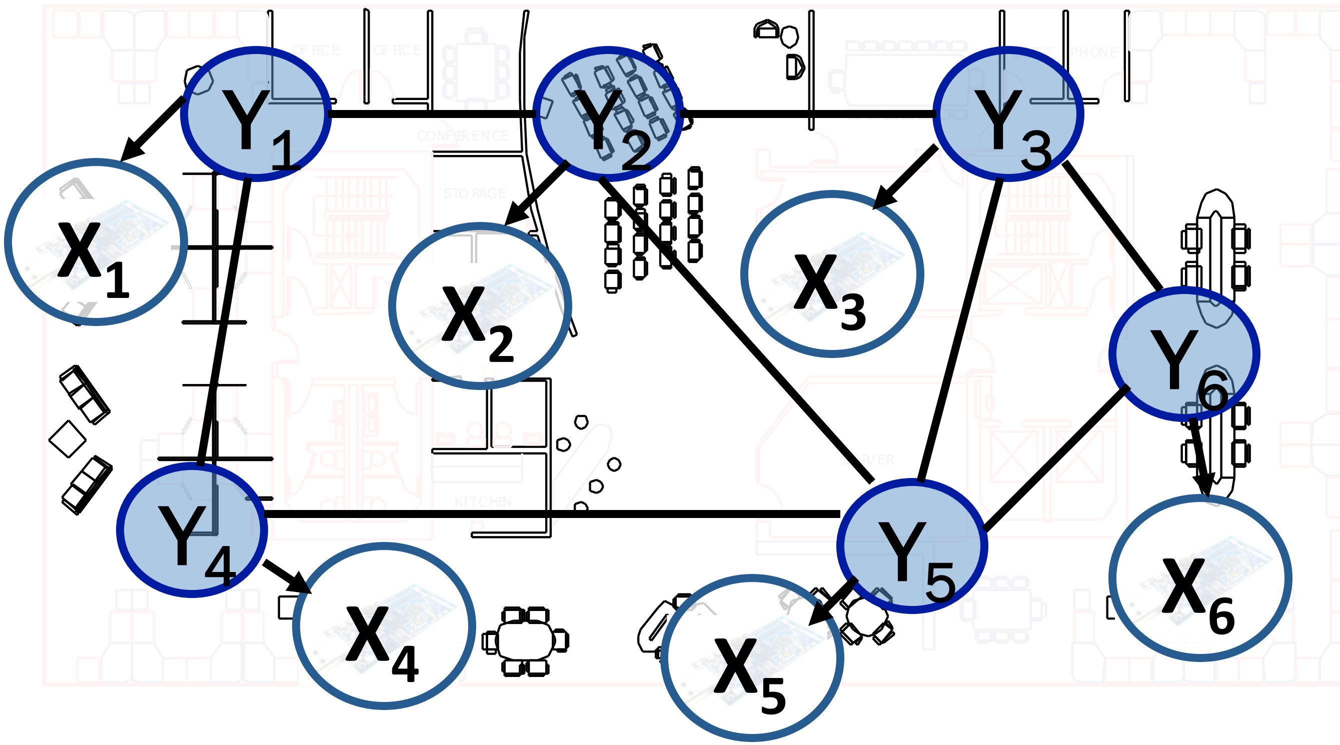
$$F(S) = \left| \bigcup_{v \in S} \mathrm{area}(v) \right|$$





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

More complex model for sensing



Y_s: temperature at location s

X_s: sensor value at location s

$X_s = Y_s + \text{noise}$

Joint probability distribution

$$P(X_1,\ldots,X_n,Y_1,\ldots,Y_n) = \underbrace{P(Y_1,\ldots,Y_n)}_{\textbf{Prior}}\ \underbrace{P(X_1,\ldots,X_n \mid Y_1,\ldots,Y_n)}_{\textbf{Likelihood}}$$

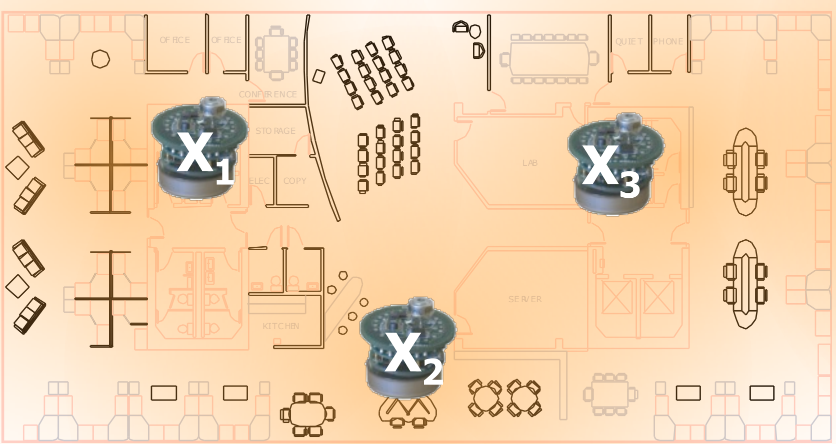
Sensor placement

Utility of having sensors at subset A of all locations

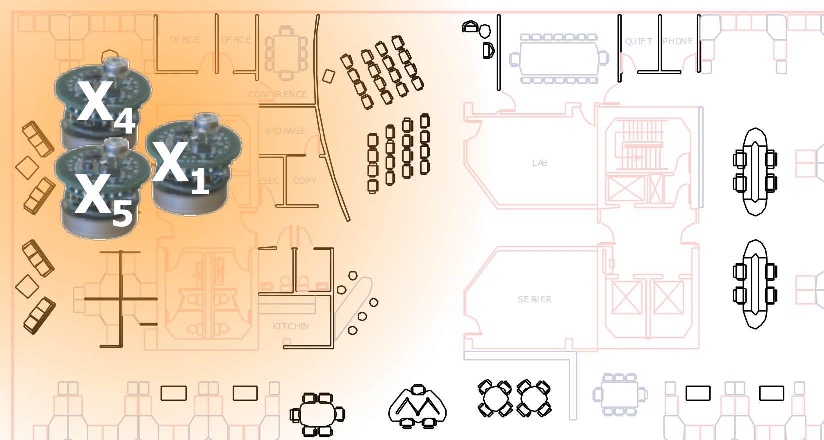
$$F(A) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X}_A) \quad = I(\mathbf{Y}; \mathbf{X}_A)$$

Uncertainty about temperature Y **before** sensing

Uncertainty about temperature Y **after** sensing



A={1,2,3}: High value F(A)



A={1,4,5}: Low value F(A)

Information gain

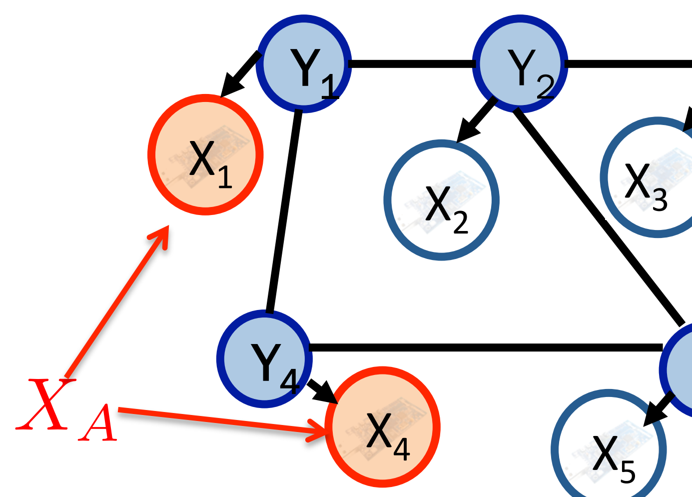
$X_1, \ldots X_n, Y_1, \ldots, Y_m$ discrete random variables

$$F(A) = I(Y; X_A) \quad = H(X_A) - H(X_A | Y)$$

modular!

$$= \sum_{i \in A} H(X_i | Y)$$

if all X_i, X_j conditionally independent given Y then *F* is submodular!



Entropy

$X_1, \ldots, X_n$ discrete random variables: $X_e \in \{1, \ldots, m\}$

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

$$H(X_e) = \sum_{x \in \{1,\ldots,m\}} P(X_e = x) \log P(X_e = x)$$

$$\boxed{A \subset B, e \notin B \qquad F(A \cup e) - F(A) \geq F(B \cup e) - F(B)??}$$

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

discrete entropy is submodular!

Submodularity and independence

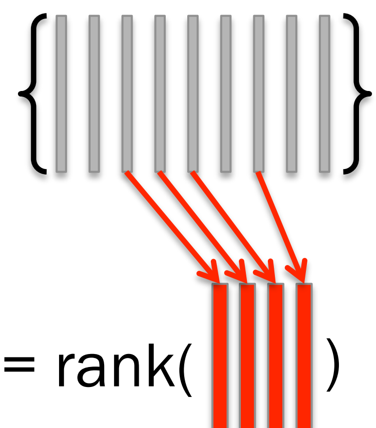
$X_1, \ldots, X_n$ discrete random variables

$X_i, i \in S$ statistically independent

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

Similarly: linear independence

$\mathcal{V} =$



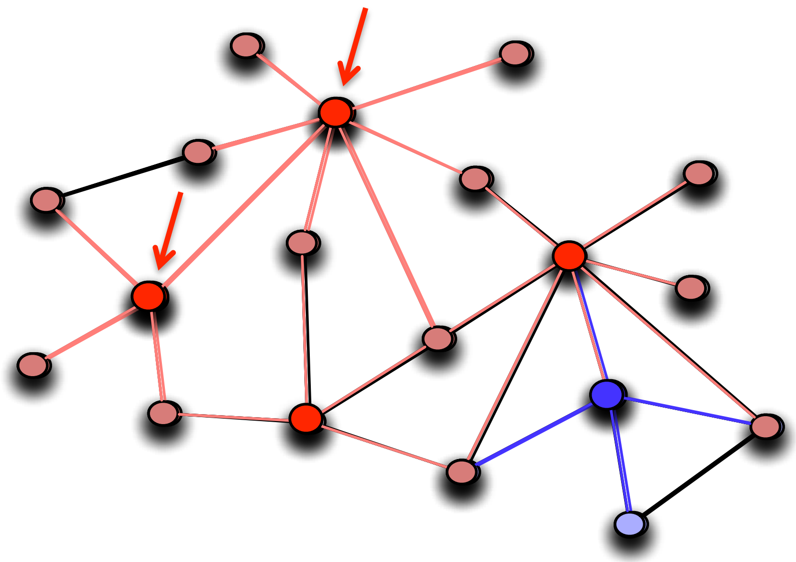
vectors in S linearly independent

⇔ F is modular/linear on S:
F(S) = |*S*|

F(S) = rank( )

Maximizing Influence

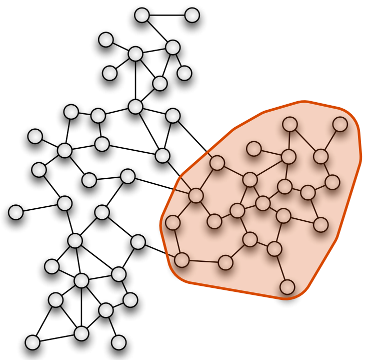
$$F(S) = \text{expected \# infected nodes}$$



$$F(S \cup s) - F(S) \geq F(T \cup s) - F(T)$$

(Kempe, Kleinberg & Tardos 2003)

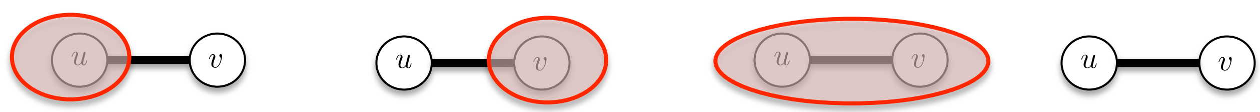
Graph cuts



- Cut for one edge:

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

$$F(\{u\}) + F(\{v\}) \geq F(\{u,v\}) + F(\emptyset)$$



w_{uv} w_{uv} 0 0

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

Useful property: sum of submodular functions is submodular

Sets and boolean vectors

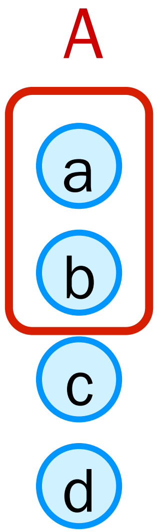
any set function

with $|V| = n$.

... is a function on binary vectors!

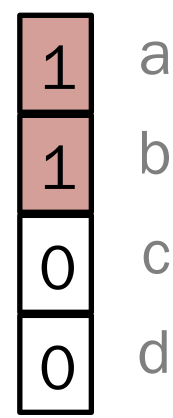
$$F : 2^V \to \mathbb{R}$$

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



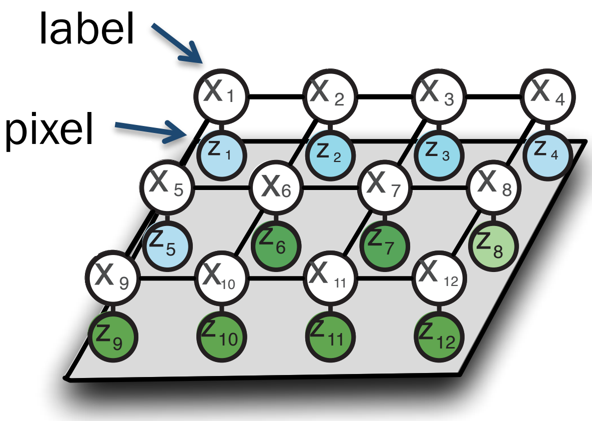
$\hat{=}$

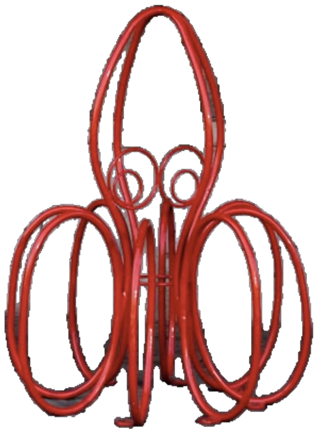
$$x = 1_A$$



subset selection = binary labeling!

Attractive potentials



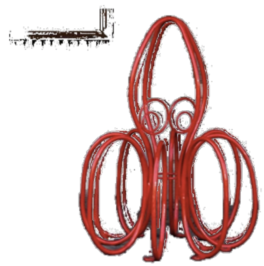


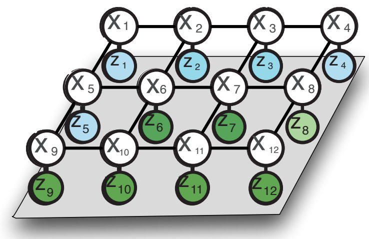
$$\max_{\mathbf{x}\in\{0,1\}^n} \quad P(\mathbf{x} \mid \mathbf{z}) \propto \exp(-E(\mathbf{x};\mathbf{z}))$$

labels — pixel values

$$\Leftrightarrow \quad \min_{\mathbf{x}\in\{0,1\}^n} \quad E(\mathbf{x};\mathbf{z})$$

Attractive potentials





$$P(\mathbf{x} \mid \mathbf{z}) \propto \exp(-E(\mathbf{x};\mathbf{z}))$$

$$E(\mathbf{x};\mathbf{z}) = \sum_i E_i(x_i) + \sum_{ij} E_{ij}(x_i, x_j)$$

spatial coherence:

$$E_{ij}(1,0) + E_{ij}(0,1) \;\geq\; E_{ij}(0,0) + E_{ij}(1,1)$$

i	j

i	j

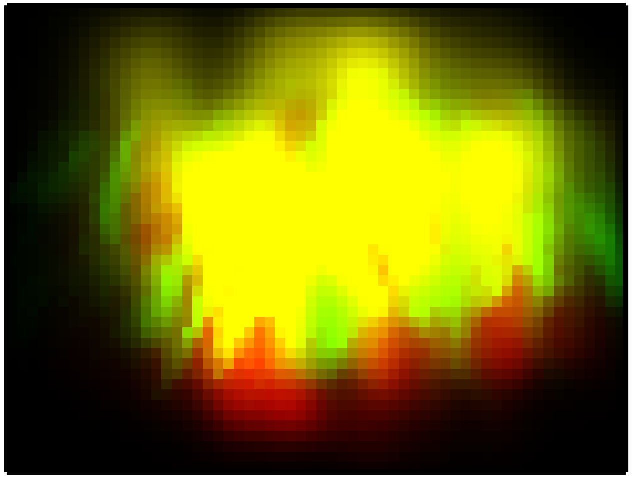
i	j

i	j

$S = \{i\}$ $\quad T = \{j\}$ $\quad S \cap T = \emptyset$ $\quad S \cup T$

$$\boxed{F(S) \;+\; F(T) \;\geq\; F(S \cup T) \;+\; F(S \cap T)}$$

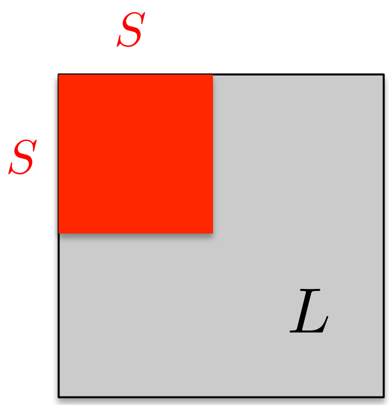
Diversity priors



$$P(S \mid \text{data}) \;\propto\; P(S)\, P(\text{data} \mid S)$$

"spread out"

Determinantal point processes



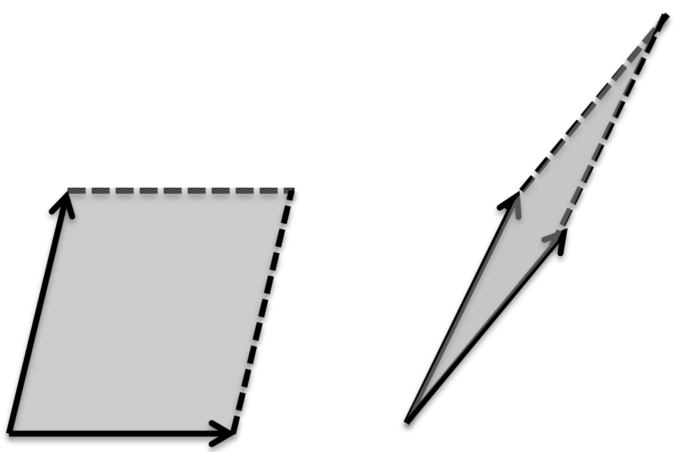
- similarity matrix L

$$L_{ij} = x_i^\top x_j$$

- sample set Y:

$$P(Y = S) \propto \det(L_S)$$

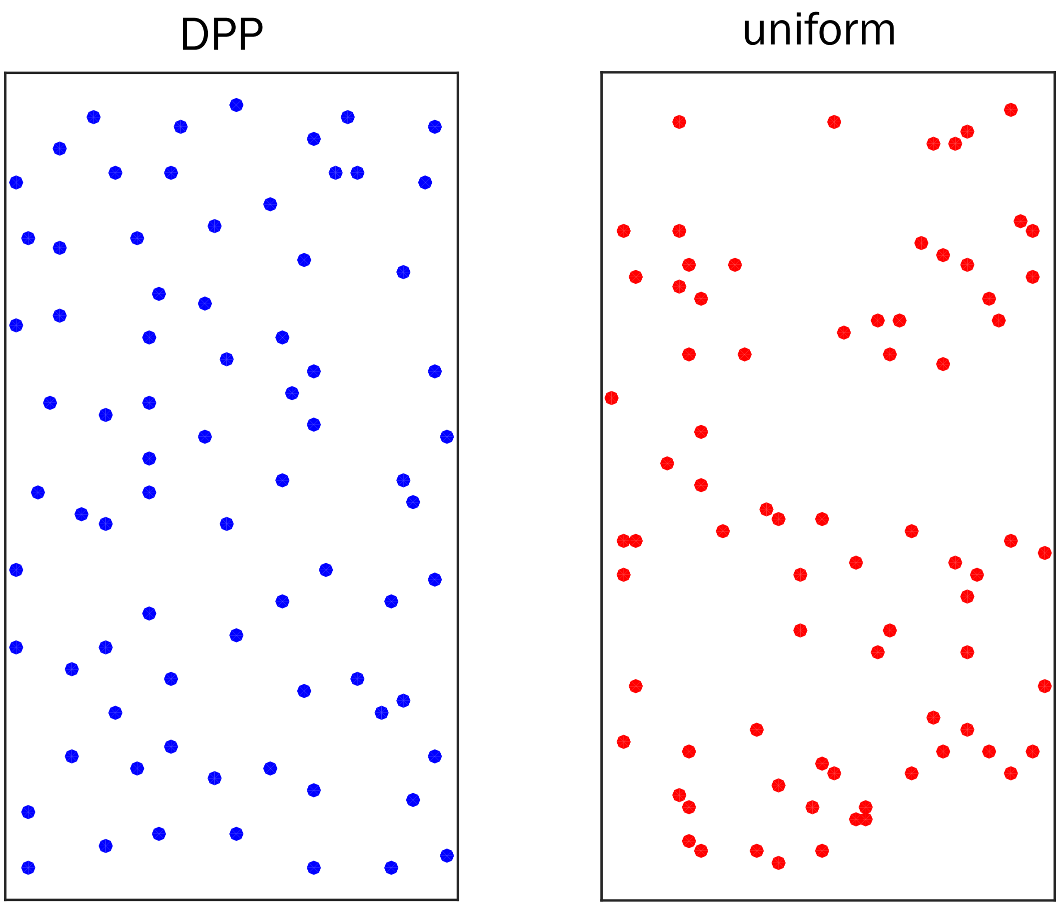
$$= \mathrm{Vol}(\{x_i\}_{i \in S})^2$$



$$F(S) = \log\det(K_S)$$

is submodular!

DPP sample



similarities:

$$s_{ij} = \exp(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2)$$

$$\sigma^2 = 35$$

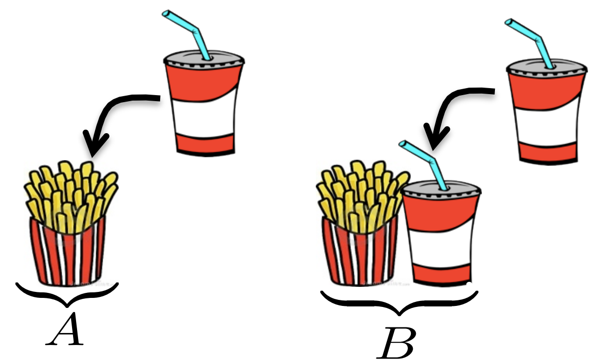
6 0 8 9 6 7 7 3 6 1 7 0 2 0 0 8 6 3 9 0 4 3 7 7 1 4 4 6 7 7

Submodularity: many examples

- linear/modular functions
- graph cut function
- coverage
- propagation/diffusion in networks

- entropy
- rank functions
- information gain

- log P(S|data) [repulsion]
  or -log P(S|data) [coherence]

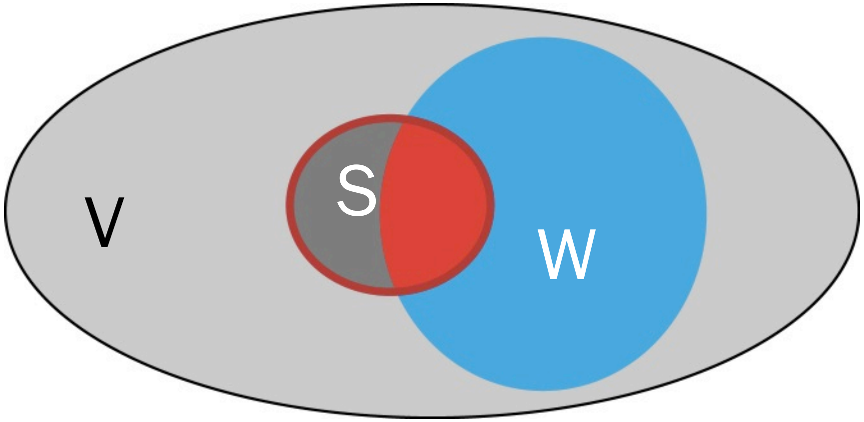


$$F(A \cup s) - F(A) \geq F(B \cup s) - F(B)$$

Closedness properties

$F(S)$ submodular on V. The following are submodular:

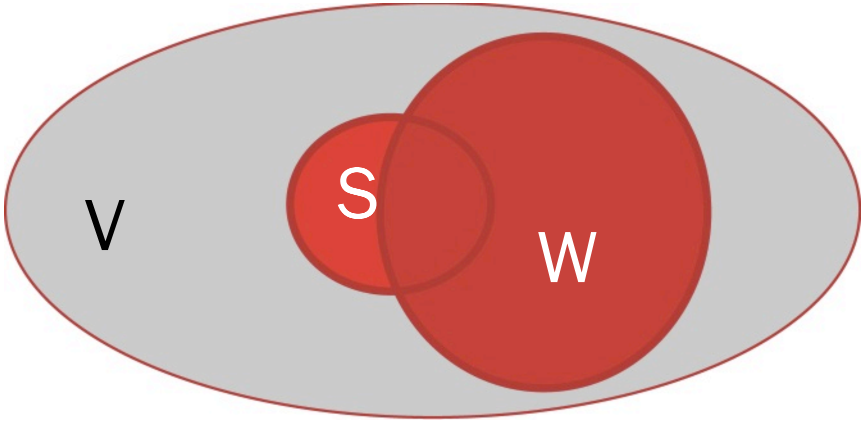
- Restriction: $F'(S) = F(S \cap W)$



Closedness properties

$F(S)$ submodular on V. The following are submodular:

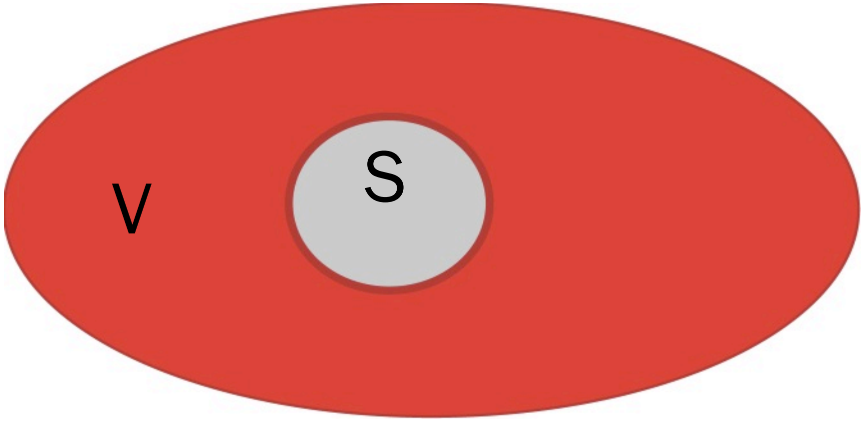
- Restriction: $F'(S) = F(S \cap W)$
- Conditioning: $F'(S) = F(S \cup W)$



Closedness properties

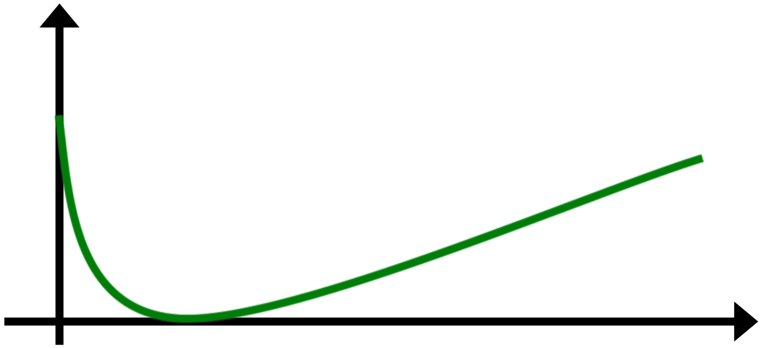
$F(S)$ submodular on V. The following are submodular:

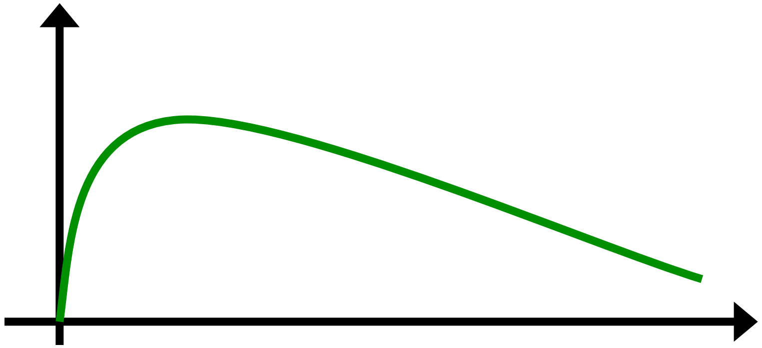
- Restriction: $F'(S) = F(S \cap W)$
- Conditioning: $F'(S) = F(S \cup W)$
- Reflection: $F'(S) = F(V \setminus S)$



Submodularity ...

discrete convexity ....





... or concavity?

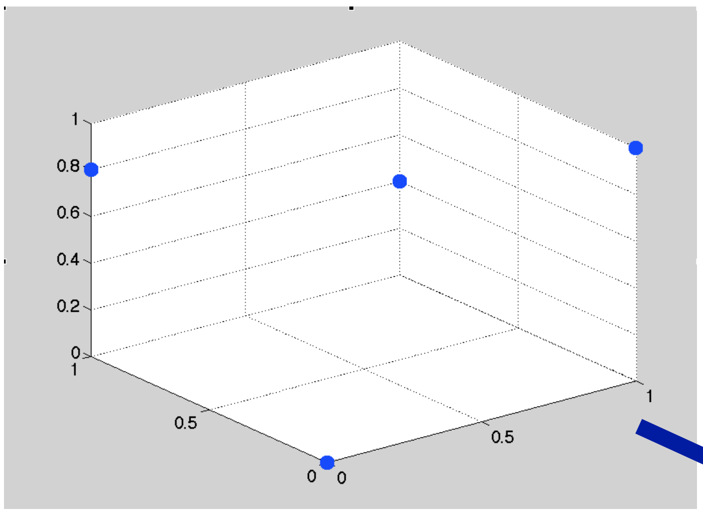
Convex functions *(Lovász, 1983)*

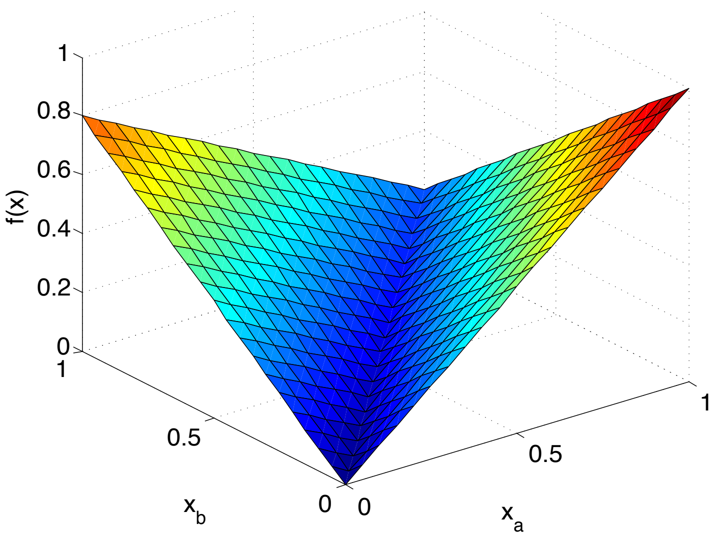
- "occur in many models in economy, engineering and other sciences", "often the only nontrivial property that can be stated in general"
- preserved under many operations and transformations: larger effective range of results
- sufficient structure for a "mathematically beautiful and practically useful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

Convex aspects

- convex extension
  - duality
  - efficient minimization





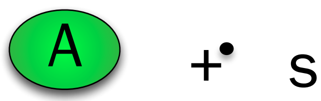
But this is only half of the story...

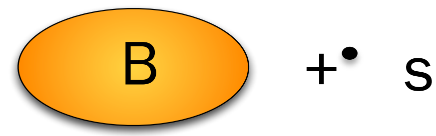
Concave aspects

- submodularity:

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

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

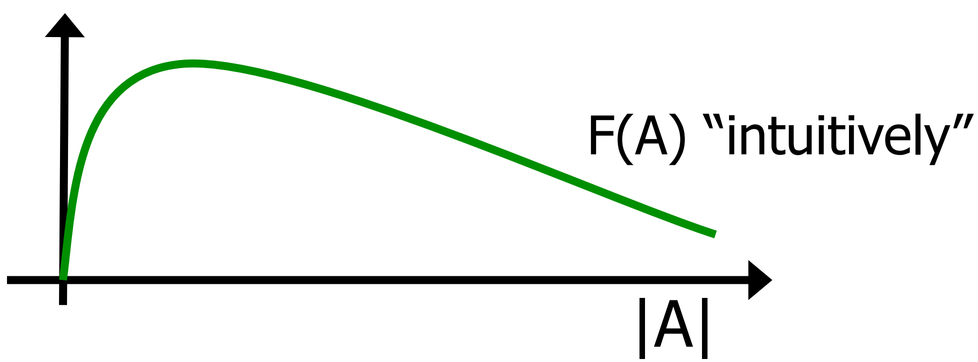




- concavity:

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

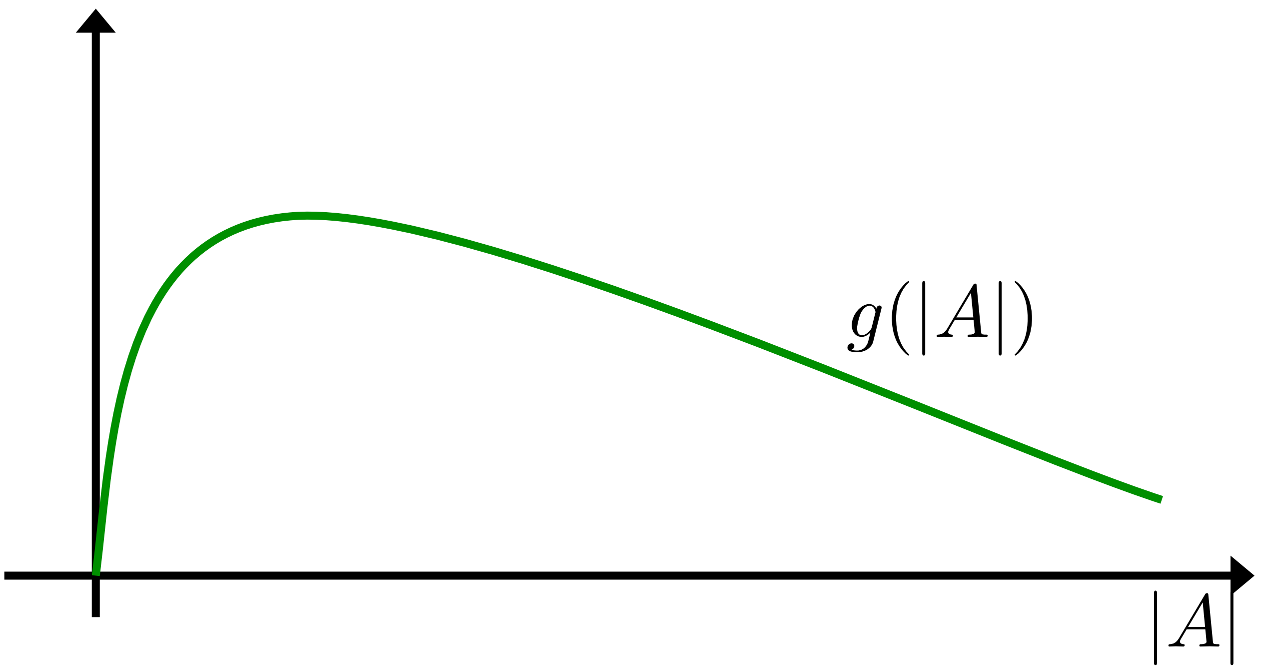
$$f(a+s) - f(a) \quad \geq \quad f(b+s) - f(b)$$



Submodularity and concavity

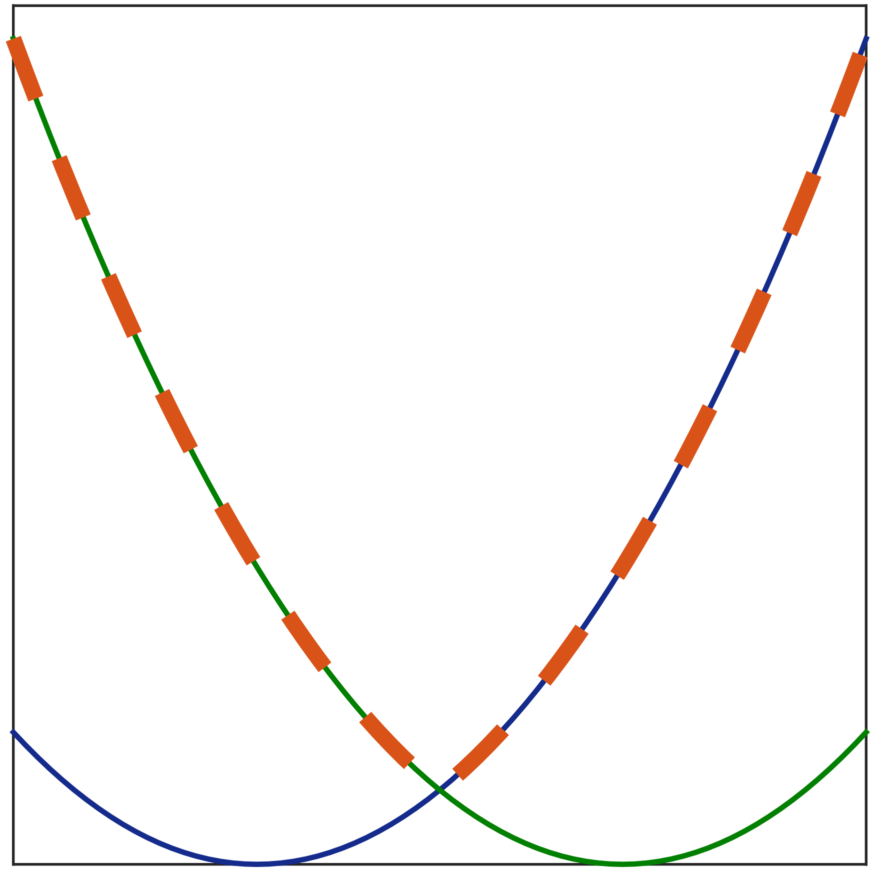
- suppose $g : \mathbb{N} \rightarrow \mathbb{R}$ and $F(A) = g(|A|)$

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



Max / min

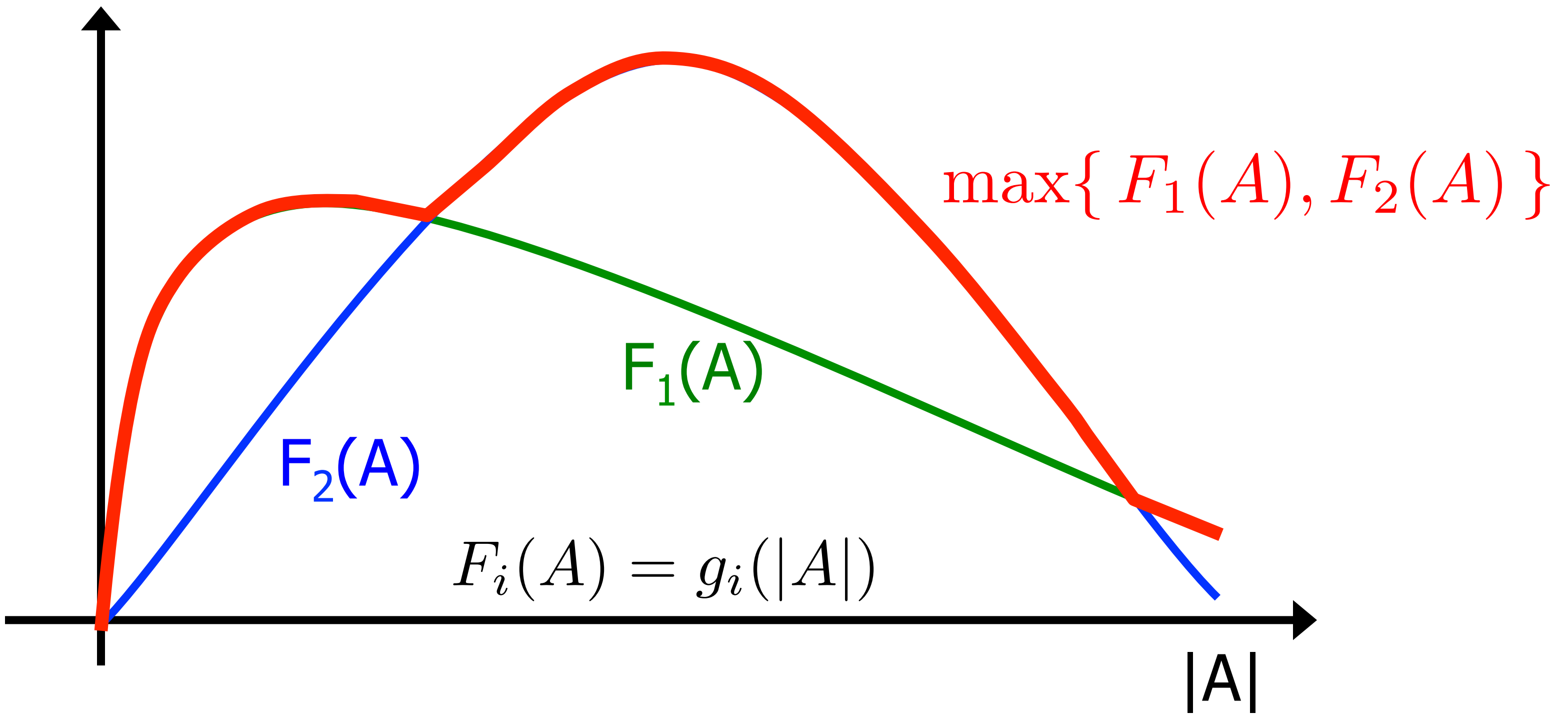
- Maximum of convex functions is convex



Maximum of submodular functions

- $F_1(A), F_2(A)$ submodular. What about

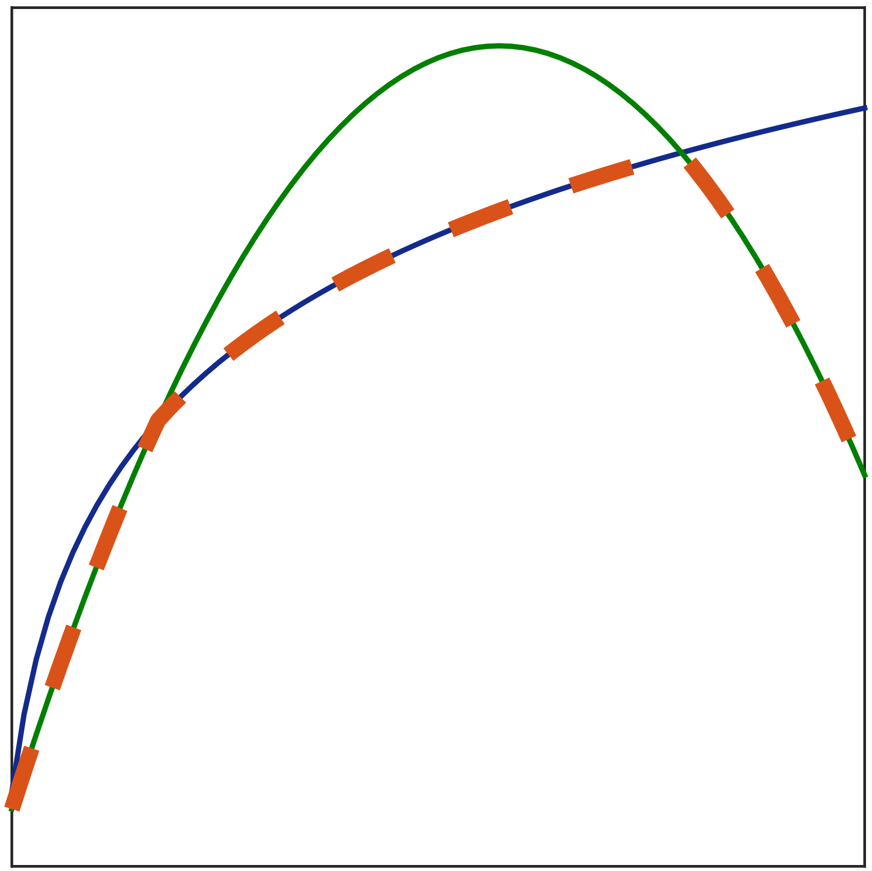
$$F(A) = \max\{\, F_1(A), F_2(A) \,\} \text{ ?}$$



$\max\{\, F_1, F_2 \,\}$ not submodular in general!

Max / min

- Minimum of concave functions is concave



Minimum of submodular functions

What about $F(A) = \min\{ F_1(A), F_2(A) \}$?

$$\overset{0}{F(A)} + \overset{0}{F(B)} \geq \overset{1}{F(A \cup B)} + \overset{0}{F(A \cap B)} \text{ ?}$$

		$F_1(A)$	$F_2(A)$
$A \cap B$	{}	0	0
A	{a}	1	0
B	{b}	0	1
$A \cup B$	{a,b}	1	1

$A \cap B$

A

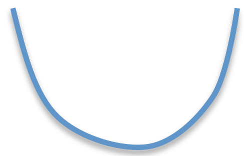
B

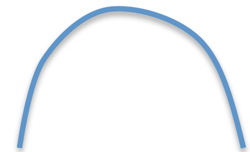
$A \cup B$

min(F_1,F_2) not submodular in general!

Submodular optimization

convex …



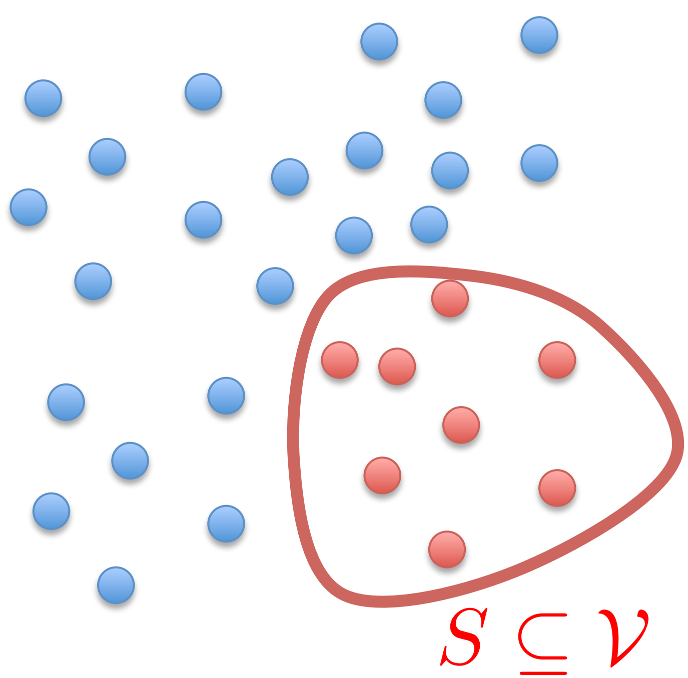


… and concave aspects!

- Maximizing submodular functions:
  diversity, repulsion, concavity
  greed is not too bad

- Minimizing submodular functions:
  coherence, regularization, convexity
  magic with polytopes, and “discrete analog of convex”

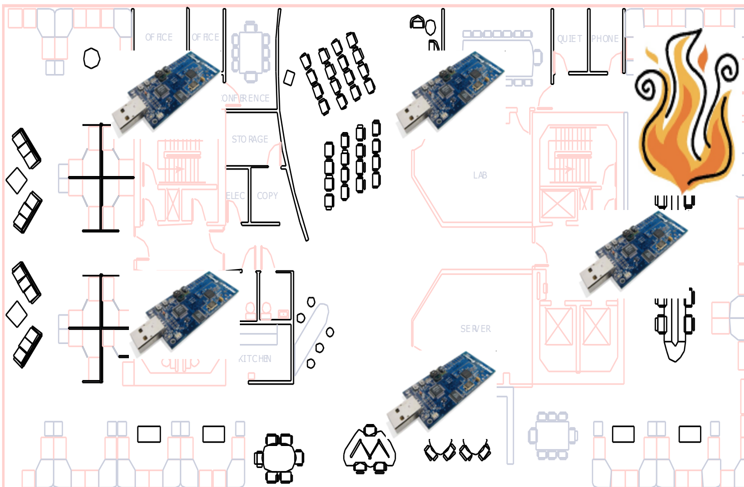
Submodular Maximization



- ground set $\mathcal{V}$
- (scoring) function $F : 2^{\mathcal{V}} \to \mathbb{R}_+$

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

Informative Subsets

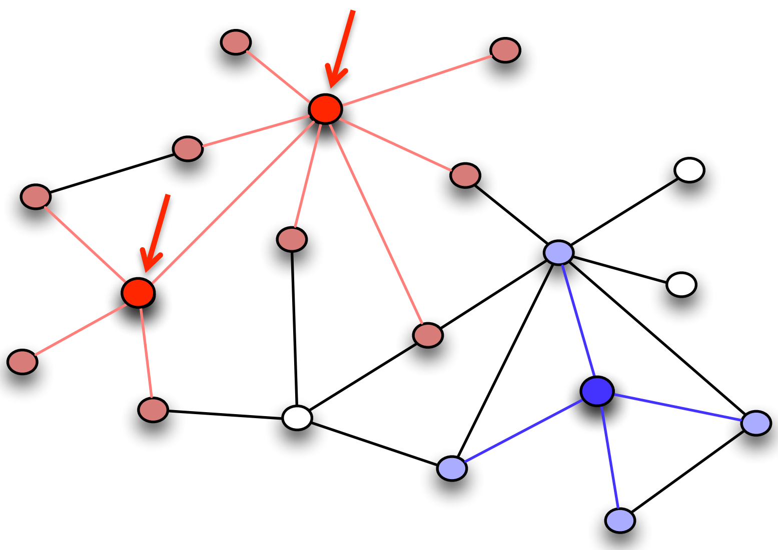


- where put sensors?
- which experiments?
- summarization

$$F(S) = \text{“information”}$$

Maximizing Influence

$$F(S) = \text{expected } \# \text{ infected nodes}$$



Kempe, Kleinberg & Tardos 2003

Summarization

- videos, text, pictures ...
- would like:
  relevance, reliability, diversity

Summarization

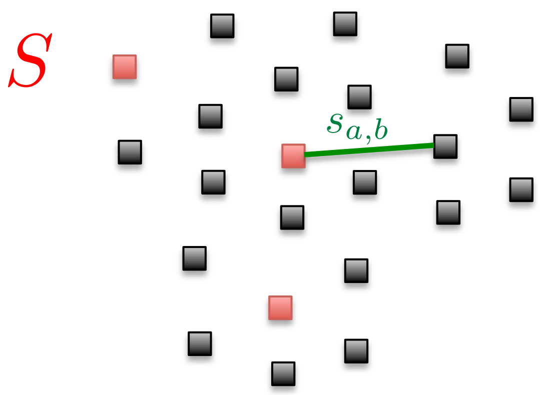
$$F(S) = R(S) + D(S)$$

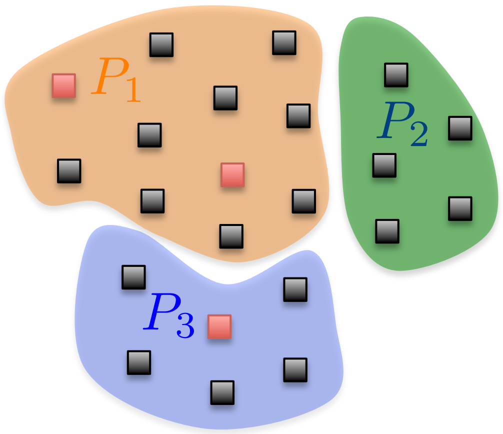
- Coverage / relevance

$$R(S) = \sum_{a \in \mathcal{V}} \max_{b \in S} s_{a,b}$$

- Diversity

$$D(S) = \sum_{j=1}^{m} \sqrt{|S \cap P_j|}$$



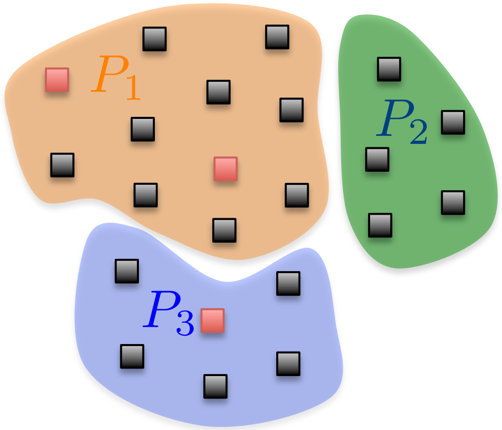


(Simon et al 2007, Lin & Bilmes 2011&2012, Tschiatschek et al 2014, Kim et al 2014, Gygli et al 2015, …)

Diversity

- Diversity

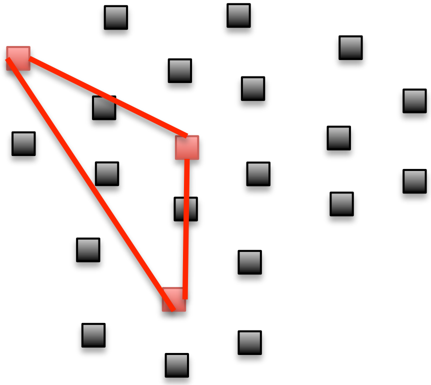
$$D(S) = \sum_{j=1}^{m} \sqrt{|S \cap P_j|}$$



increasing

Another diversity function …

$$D(S) = -\sum_{a,b \in S} s_{a,b}$$



decreasing

Summarization: results

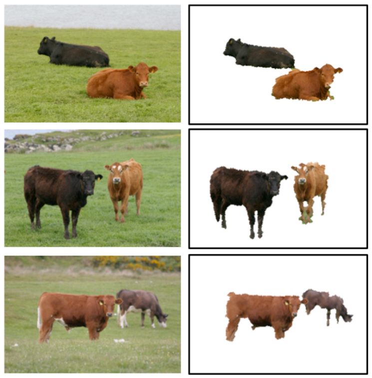
	R	F
$\mathcal{L}_1(S) + \lambda \mathcal{R}_Q(S)$	12.18	12.13
$\mathcal{L}_1(S) + \sum_{\kappa=1}^{3} \lambda_\kappa \mathcal{R}_{Q,\kappa}(S)$	**12.38**	**12.33**
Toutanova et al. (2007)	11.89	11.89
Haghighi and Vanderwende (2009)	11.80	-
Celikyilmaz and Hakkani-tür (2010)	11.40	-
Best system in DUC-07 (peer 15), using web search	**12.45**	12.29

(Lin & Bilmes 2011)

Many more functions are possible ...
➔Learn a weighted combination: structured prediction works even better!

(Lin & Bilmes 2012, Tschiatschek et al 2014, Gygli et al 2015, Xu et al 2015,...)

More maximization …

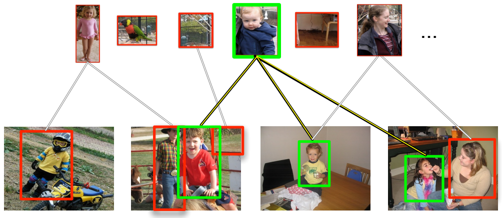


co-segmentation
by maximizing
anisotropic diffusion
(Kim et al 2011)

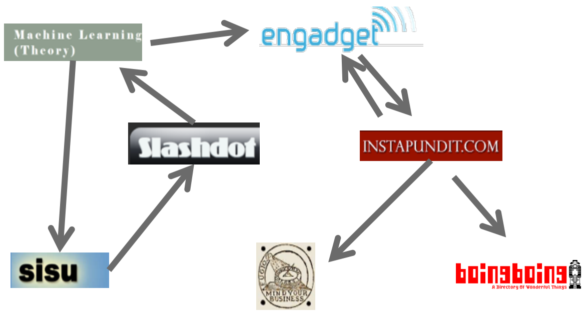
environmental monitoring
(Krause, …)

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

weakly supervised
object detection
(Song et al 2014)



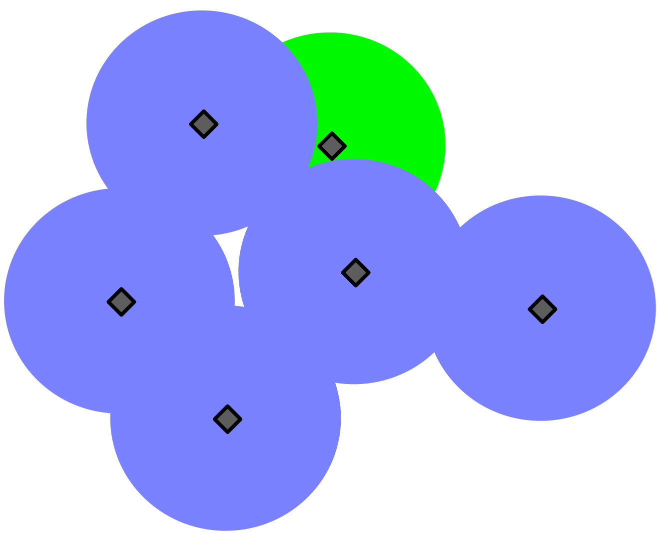
diverse
recommendations
(Yue & Guestrin)

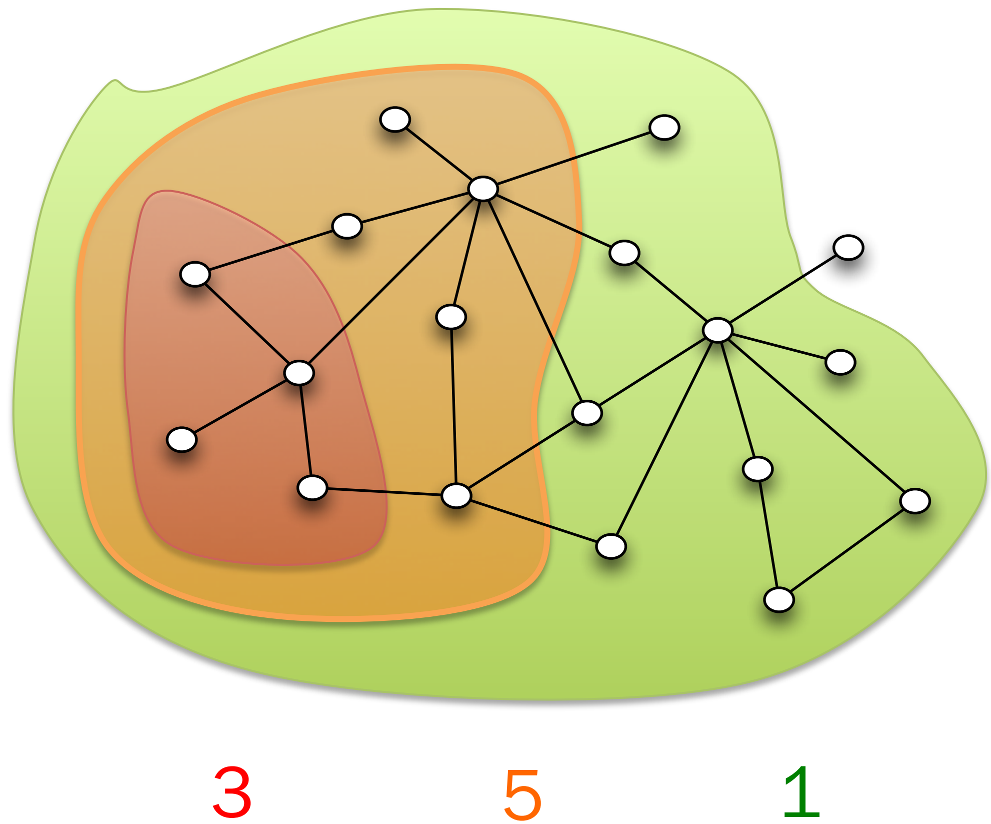


inferring networks
(Gomez Rodriguez et al 2012)

Monotonicity

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

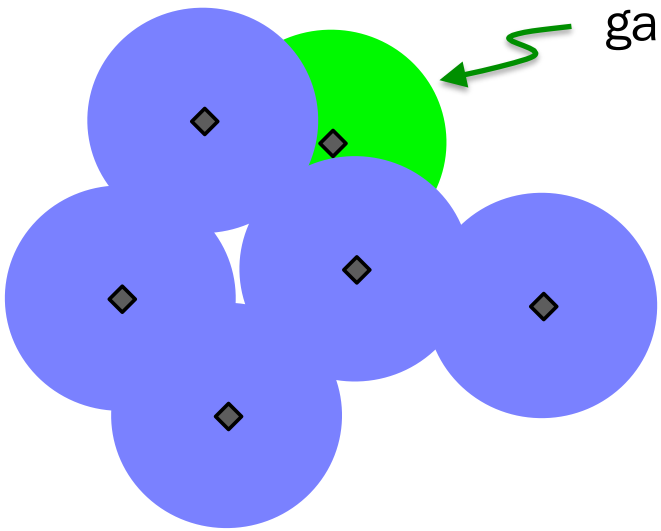




Monotonicity – how check?

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

Let $B = A \cup \{a\}$. $\qquad \underbrace{F(A \cup \{a\}) - F(A)}_{\text{marginal gain}} \geq 0.$



gain: +5 - 8

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

Maximizing monotone functions

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

$$\max \; F(S)$$

- NP-hard
- approximation: greedy algorithms

Maximizing monotone functions

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

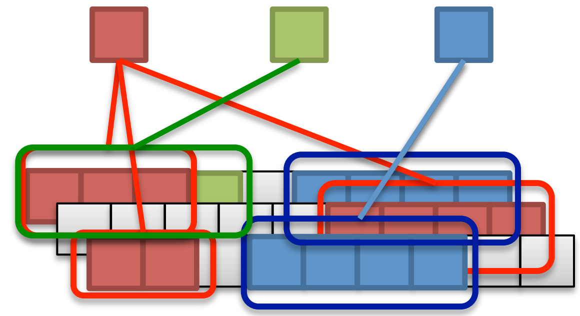
- greedy algorithm:

$$S_0 = \emptyset$$

for i = 0, ..., k-1

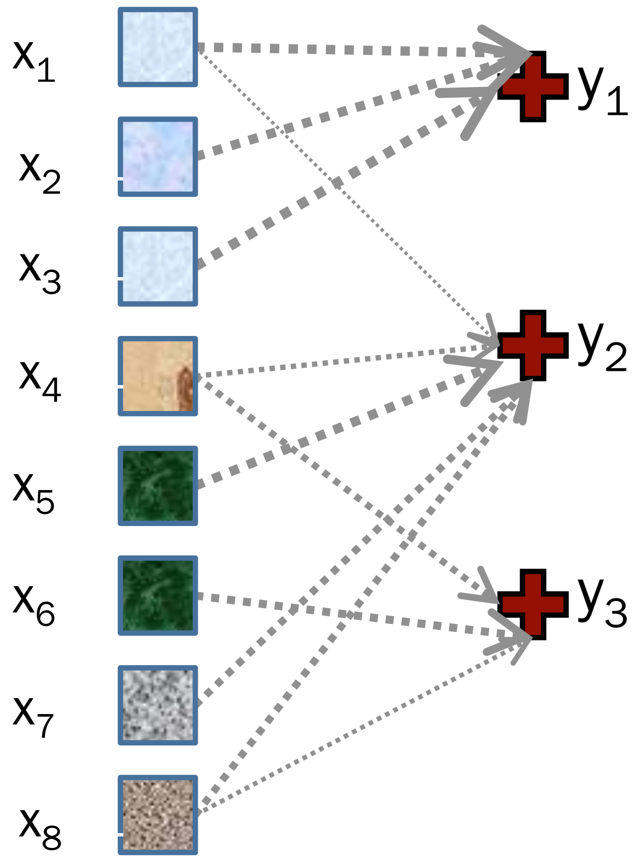
$$e^* = \arg\max_{e \in \mathcal{V} \setminus S_i} F(S_i \cup \{e\})$$

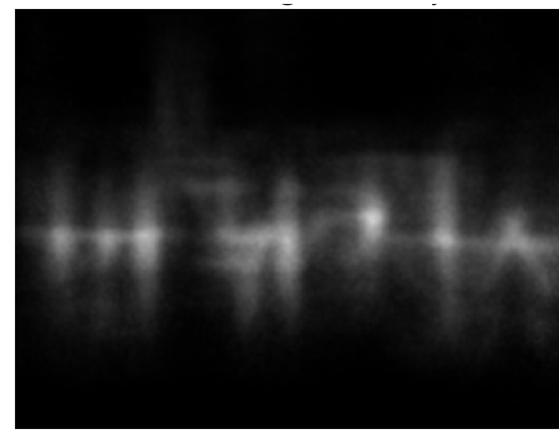
$$S_{i+1} = S_i \cup \{e^*\}$$



How "good" is S_k ?

Pedestrian detection





x_j = index of hypothesis explaining x_j

Voting elements

Hypotheses

y_i = 1: object i present

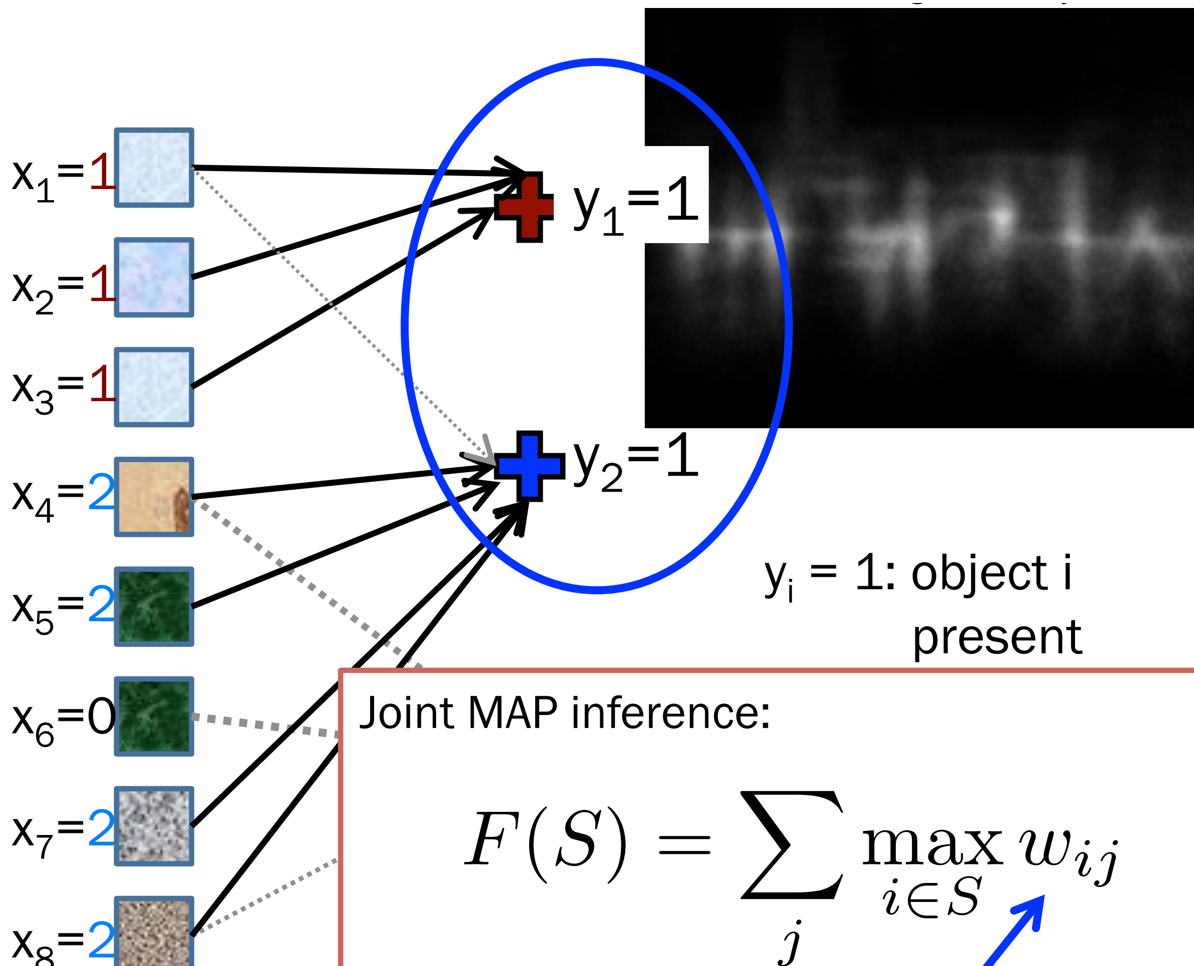
y_i = 0: object i not present

Illustrations courtesy of Pushmeet Kohli

(Barinova et al.'10)

Pedestrian detection

x_j = index of hypothesis explaining x_j



y_i = 1: object i present

Joint MAP inference:

$$F(S) = \sum_j \max_{i \in S} w_{ij}$$

Weight of element x_j wrt hyp. y_i

Voting elements

Illustrations courtesy of Pushmeet Kohli

Inference

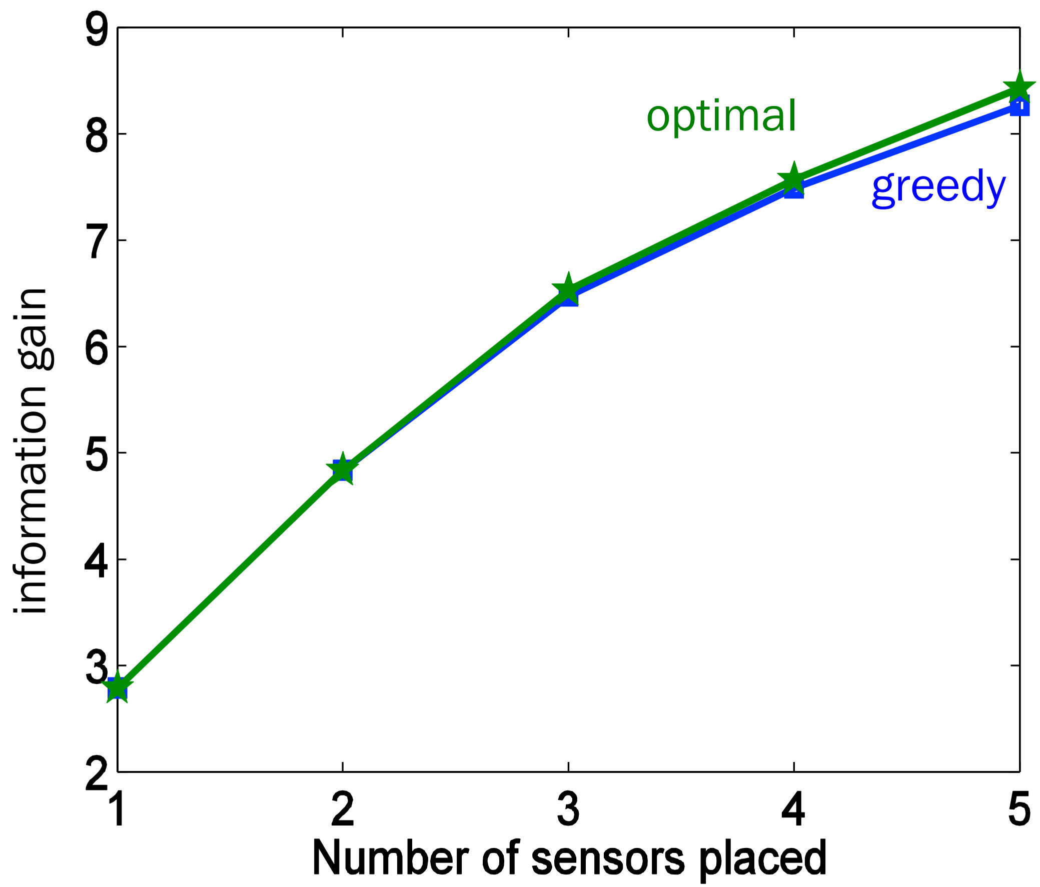
Datasets from [Andriluka et al. CVPR 2008]
(with strongly occluded pedestrians added)

Using the Hough forest trained in [Gall&Lempitsky CVPR09]

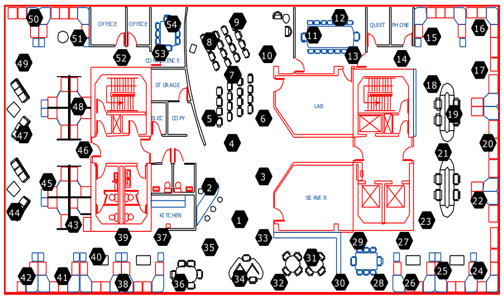
Illustrations courtesy of Pushmeet Kohli

How good is greedy? in practice...

empirically:



sensor placement



How good is greedy? ... in theory

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

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

F monotone submodular, S_k solution of greedy. Then

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

optimal solution

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

Questions

- What if I have more complex constraints?
  - budget constraints
  - matroid constraints

- Greedy takes *O(nk)* time. What if n, k are large?

- What if my function is not monotone?

More complex constraints: budget

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

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

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

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

➔ approximation factor:

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

even better but less fast: partial enumeration *(Sviridenko, 2004)* or filtering *(Badanidiyuru & Vondrák 2014)*

(Leskovec et al 2007)

Other constraints: Camera network

- Ground set: $V = \{1_a, 1_b, \ldots, 5_a, 5_b\}$
- Sensing quality model: $F : 2^V \to \mathbb{R}$

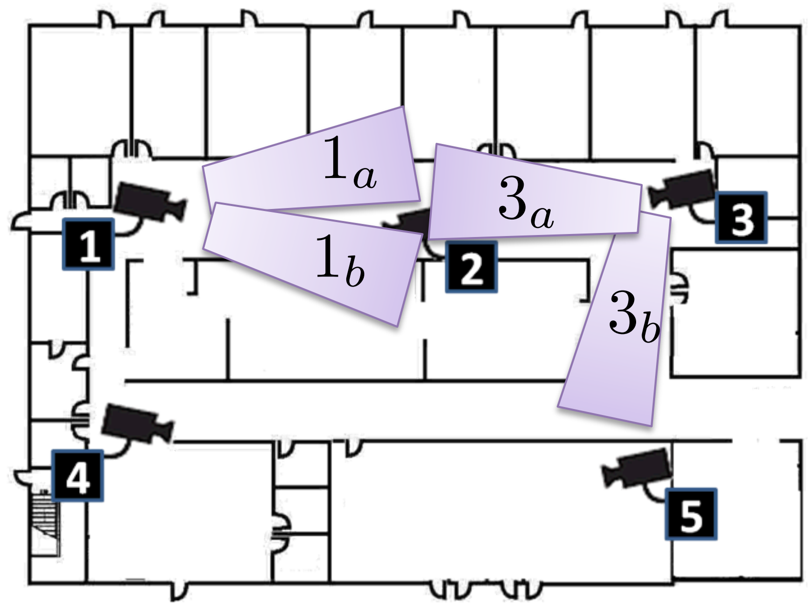
- Configuration (subset) is feasible if no camera is pointed in two directions at once

- Constraints:

$$P_1 = \{1_a, 1_b\}, \ldots, P_5 = \{5_a, 5_b\}$$

require:

$$|S \cap P_i| \leq 1$$



Generalization of Greedy algorithm

$$S = \emptyset$$

While $\exists e : S \cup e$ feasible

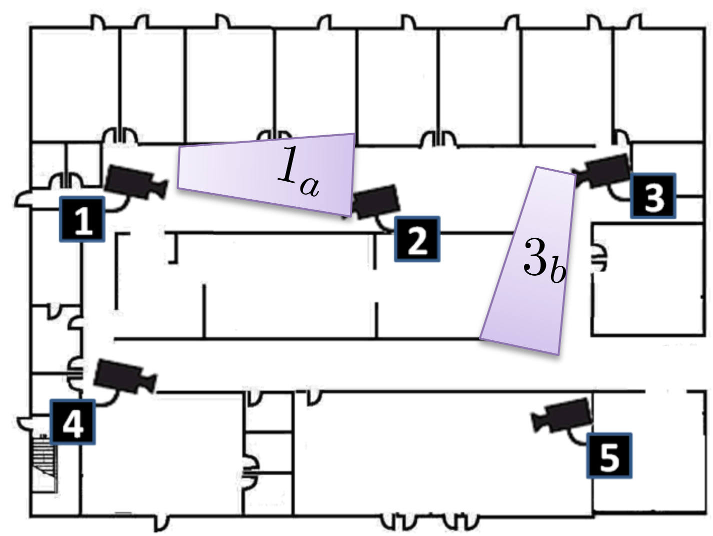
$$e^* \leftarrow \text{argmax}\{F(S \cup e) \mid S \cup e \text{ feasible}\}$$

$$S \leftarrow S \cup e^*$$

Theorem *(Nemhauser, Wolsey, Fisher 78)*

For monotone submodular functions:

$$F(S_{\text{greedy}}) \geq \frac{1}{2} F(S^*)$$



- Does this always work?

No. But works for matroid constraints.

Matroids: examples

set S is independent (= feasible) if …

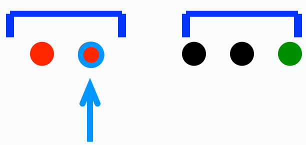
… |S| ≤ k

Uniform matroid

- S independent ➔ $T \subseteq S$ also independent

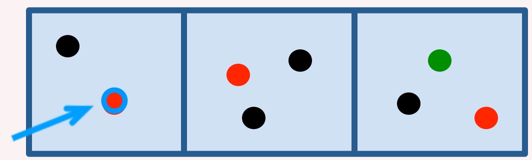
Matroids

set S is independent (= feasible) if …



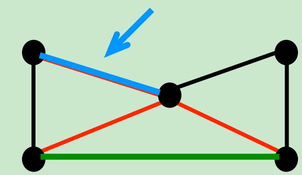
… |S| ≤ k

Uniform matroid



… S contains at most one element from each group

Partition matroid



… S contains no cycles

Graphic matroid

- S independent ➔ $T \subseteq S$ also independent
- Exchange property: S, U independent, $|S| > |U|$
  ➔ some $e \in S$ can be added to U: $U \cup e$ independent
- All maximal independent sets have the same size

Generalization of Greedy algorithm

$S = \emptyset$

While $\exists e : S \cup e$ feasible

$$e^* \leftarrow \operatorname{argmax}\{F(S \cup e) \mid S \cup e \text{ feasible}\}$$

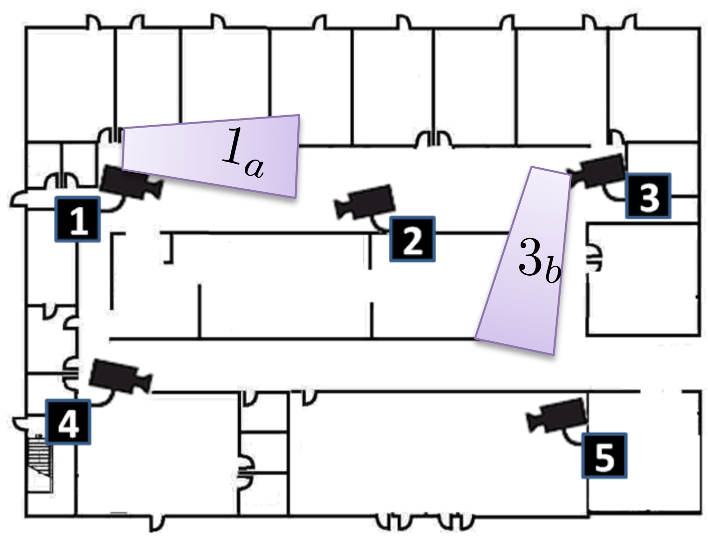
$$S \leftarrow S \cup e^*$$

Theorem *(Nemhauser, Wolsey, Fisher 78)*

For monotone submodular functions:

$$F(S_{\text{greedy}}) \geq \frac{1}{2} F(S^*)$$

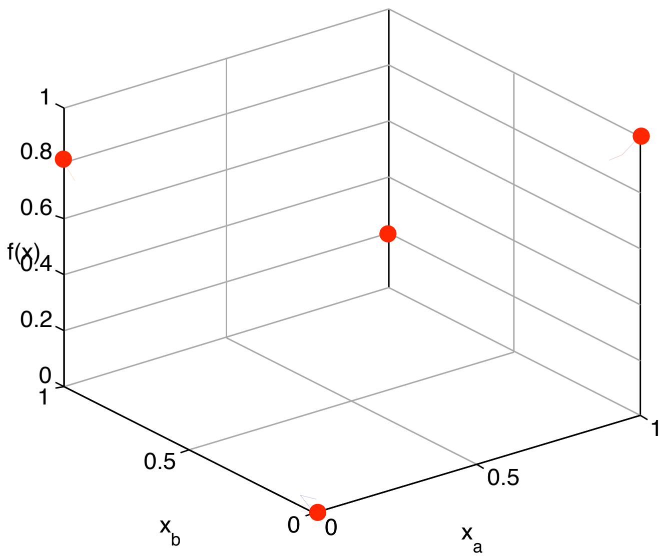
- Works for matroid constraints
- Is this the best possible?



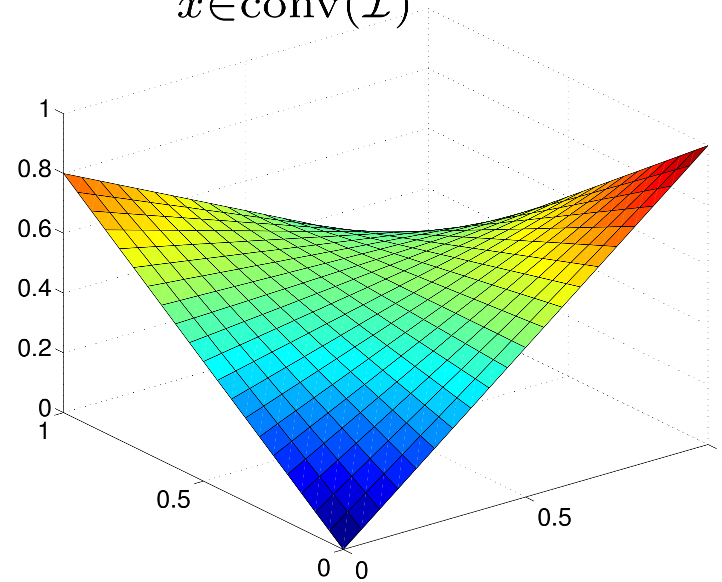
Can do a bit better with relaxation: (1-1/e)

Relax: Discrete to continuous

$$\max_{S \in \mathcal{I}} F(S)$$



$$\max_{x \in \mathrm{conv}(\mathcal{I})} f_M(x)$$



Algorithm:

1. approximately maximize f_M
   (like Frank-Wolfe algorithm – next lecture)
2. round to discrete set (pipage rounding)

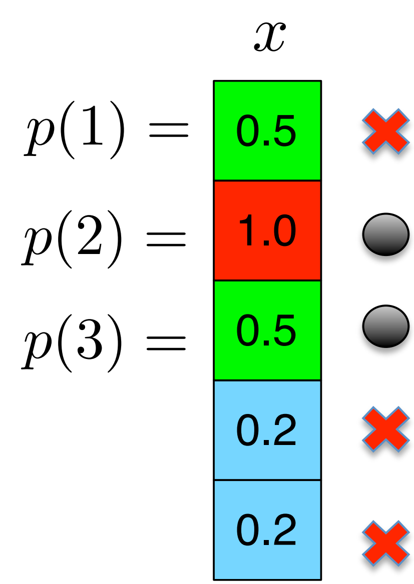
(Calinescu-Chekuri-Pal-Vondrak 2011)

Multilinear extension

- sample item e with probability x_e

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

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



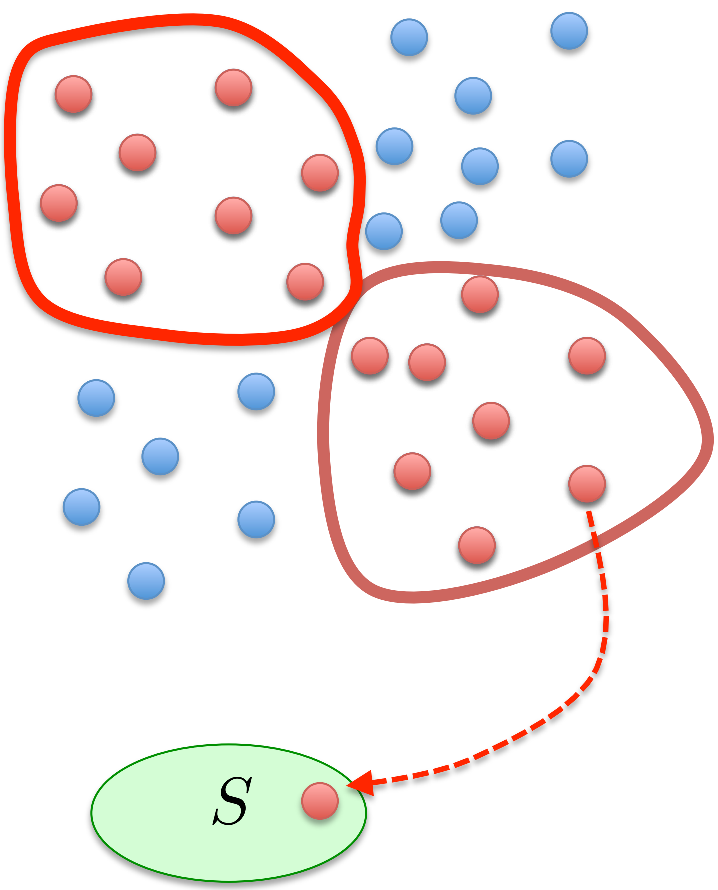
Questions

- What if I have more complex constraints?
  - budget constraints
  - matroid constraints

- Greedy takes $O(nk)$ time. What if n, k are large?
  - faster sequential algorithms
  - filtering
  - parallel / distributed

- What if my function is not monotone?

Making greedy faster: stochastic



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

for i=1…k:

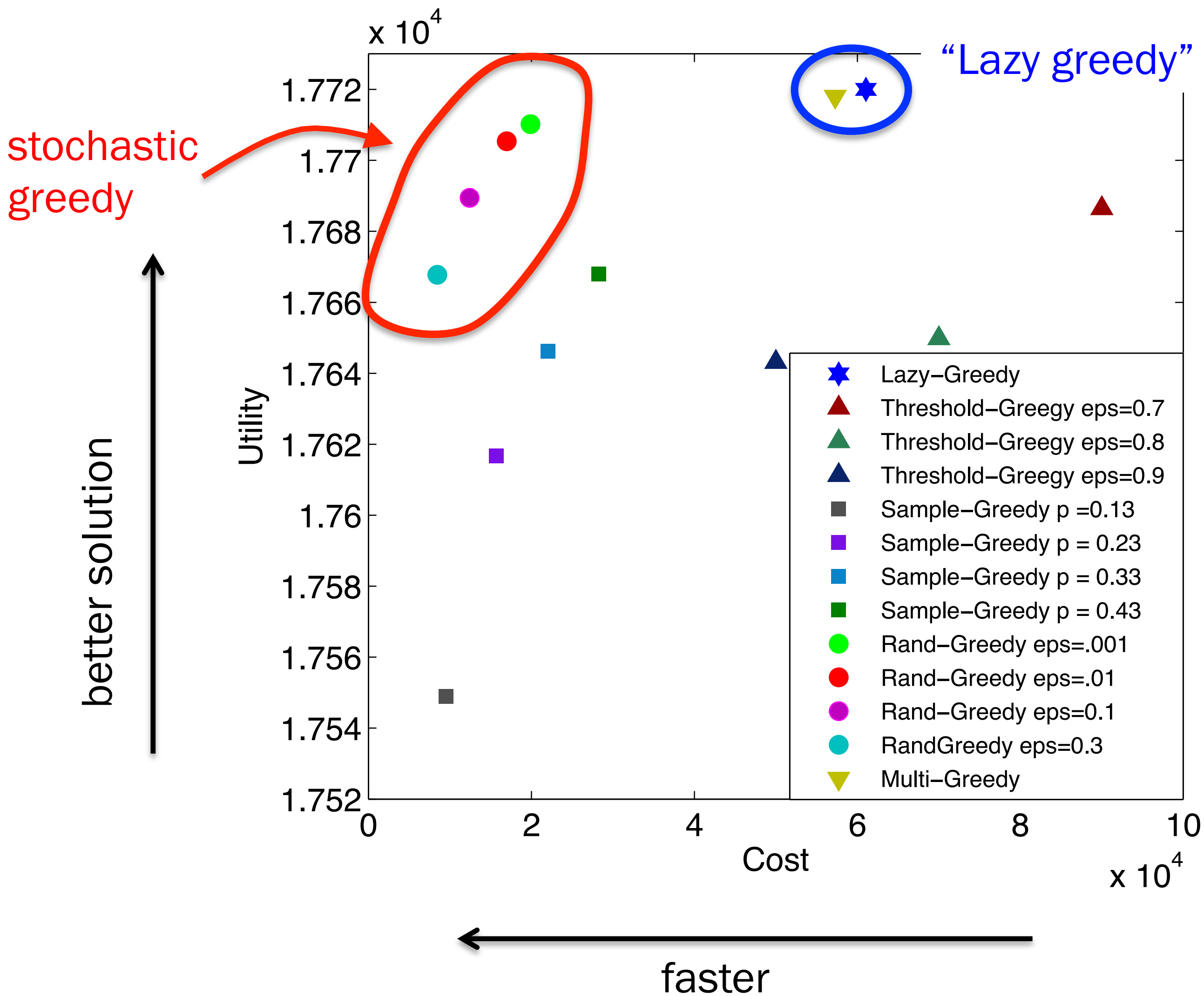
- randomly pick set *T* of size $\frac{n}{k} \log \frac{1}{\epsilon}$
- find best *a* element in *T* and add

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

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

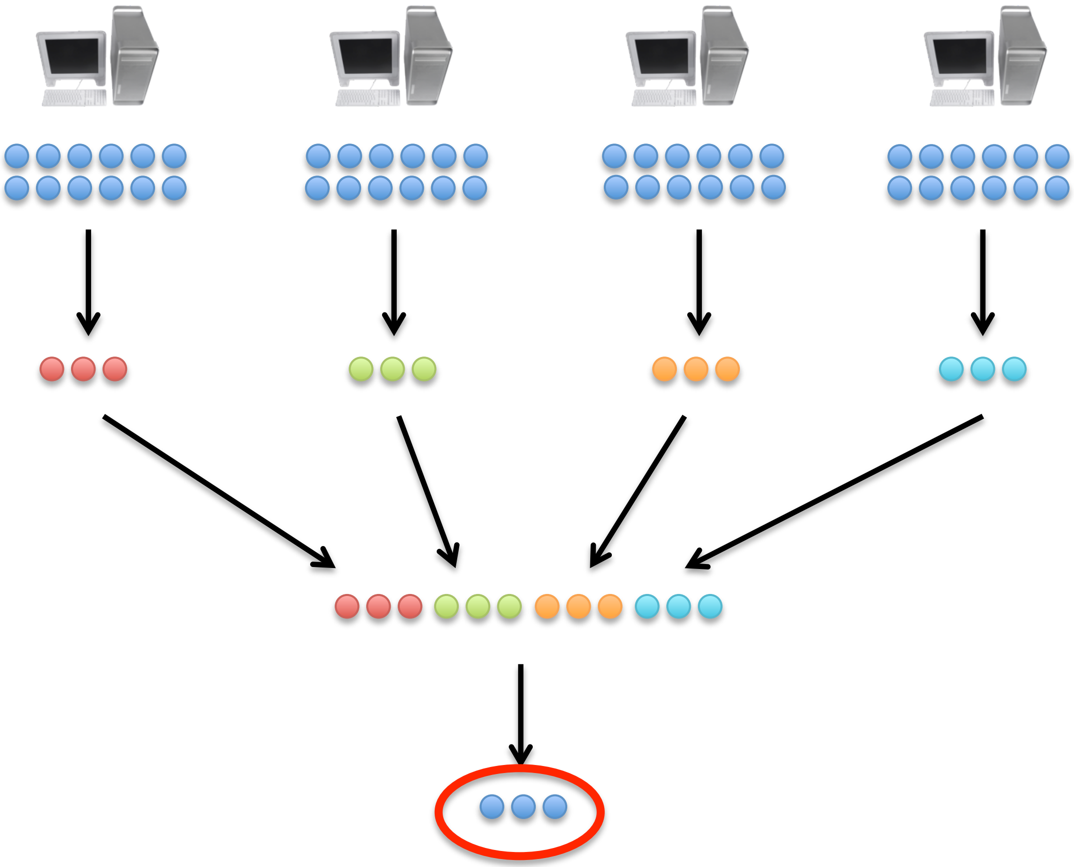
(Mirzasoleiman et al 2014)

Performance




even more data ...
distributed greedy algorithm?

Distributed greedy algorithms



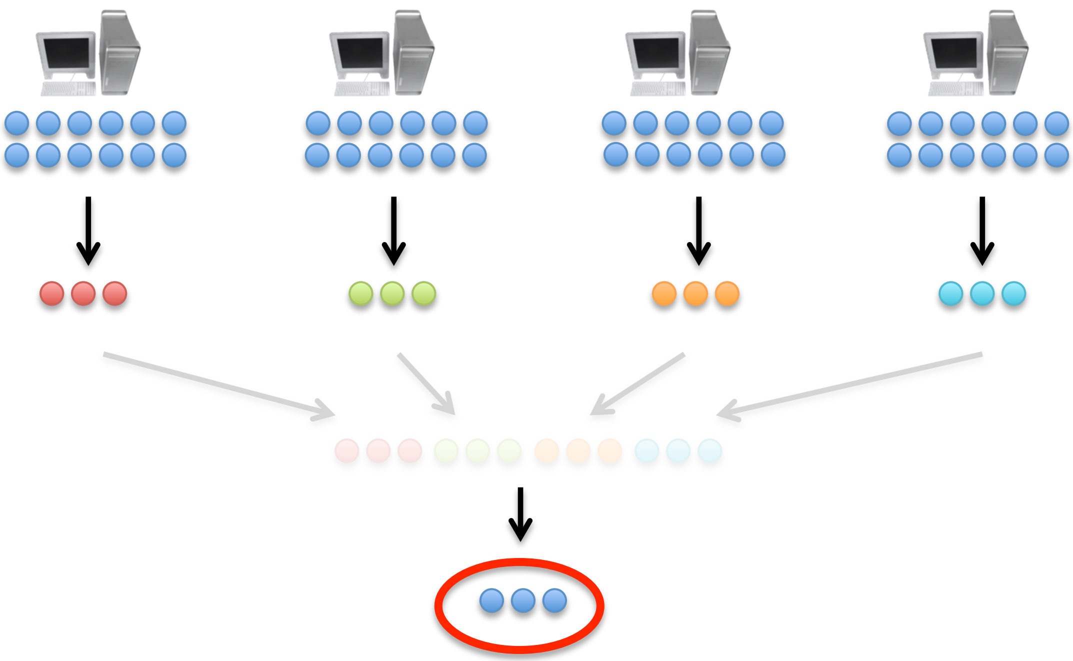
greedy is sequential.
pick in parallel??

pick *k* elements
on each machine.

combine and run
greedy again.

Is this useful?

Distributed greedy algorithms



pick in parallel
from *m* machines

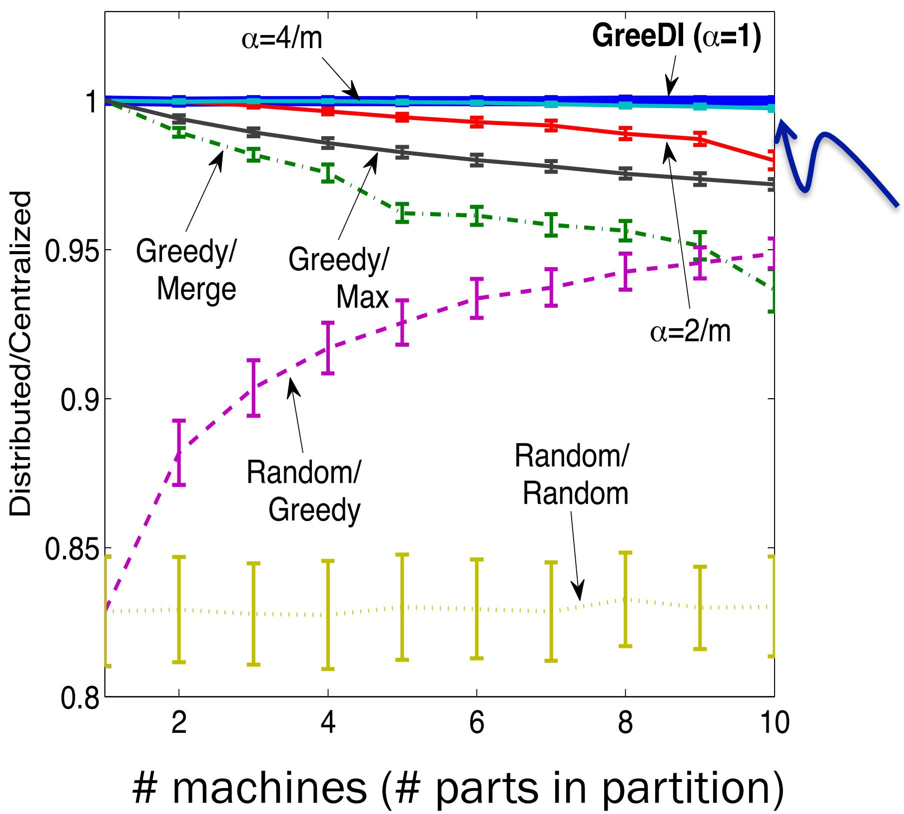
Is this useful?

Approximation factor:

$$O\Big(\frac{1}{\min\{\sqrt{k}, m\}}\Big)$$

(Mirzasoleiman et al 2013)

Distributed Greedy

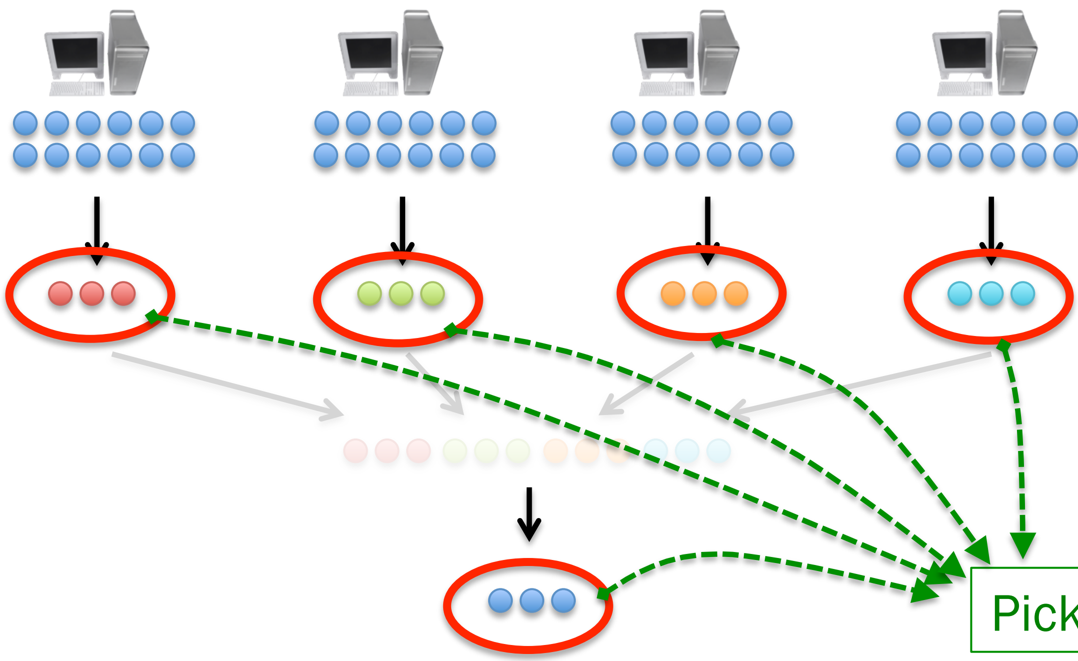


In practice, performs often quite well.

1. special structure: Improved guarantees if F is Lipschitz or a sum of many terms
2. randomization

(Mirzasoleiman et al 2013)

Distributed greedy algorithms



randomly distribute across machines

pick in parallel
from *m* machines

Pick the best of m+1 solutions

- each machine: $\alpha-$approximation algorithm
- level 2: $\beta-$ approximation algorithm

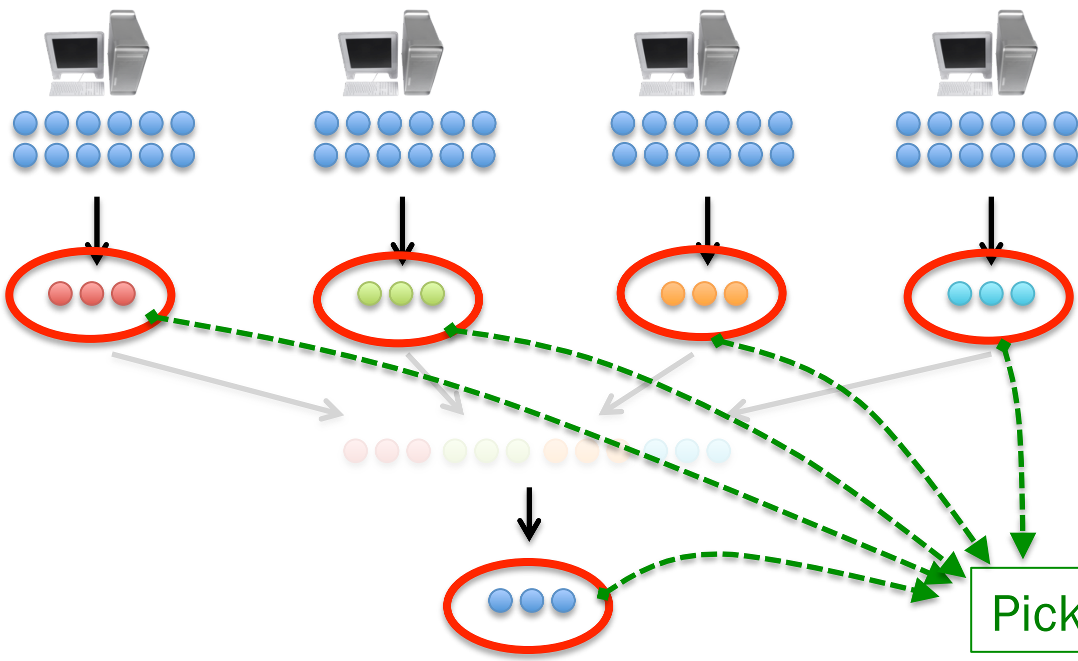
➔ overall approximation factor: $$\mathbb{E}[F(\widehat{S})] \geq \frac{\alpha\beta}{\alpha+\beta} F(S^*)$$

(Mirzasoleiman et al 2013, de Ponte Barbosa et al 2015, see also Mirrokni, Zadimoghaddam 2015)

Distributed greedy algorithms

randomly distribute across machines

pick in parallel
from *m* machines



Pick the best of m+1 solutions

$$\mathbb{E}[F(\widehat{S})] \;\geq\; \frac{\alpha\beta}{\alpha+\beta} F(S^*)$$

With greedy algorithm on both levels:
$\alpha = \beta = 1 - \frac{1}{e}$, overall factor:

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

(Mirzasoleiman et al 2013, de Ponte Barbosa et al 2015, see also Mirrokni, Zadimoghaddam 2015)

Questions

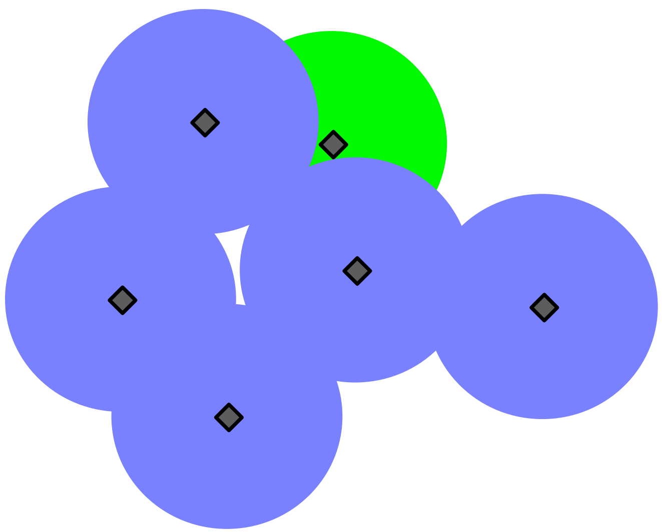
- What if I have more complex constraints?
  - matroid constraints
  - budget constraints

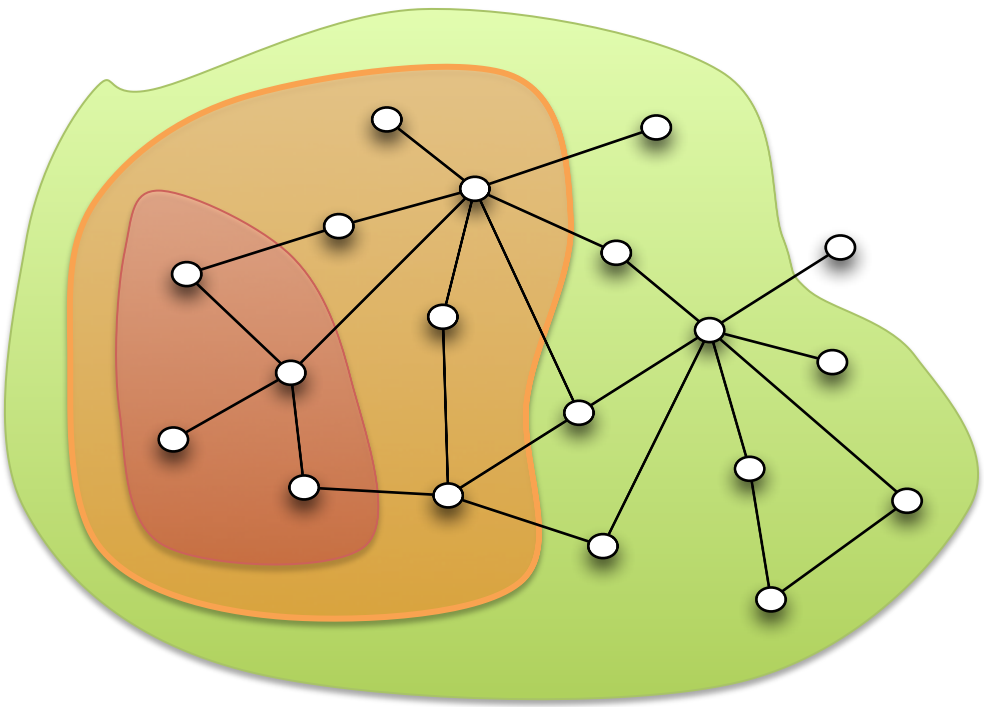
- Greedy takes $O(nk)$ time. What if n, k are large?
  - stochastic
  - parallel / distributed
  - filtering, structured, …

- 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

still assume:

$F(S) \geq 0 \quad \text{for all } S$

Greedy can fail ...

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

greed

$F(A)$

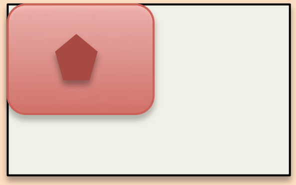
optimal solution

$F(A) = 95$

sensor 1

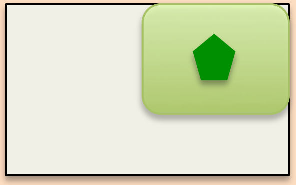
coverage: 100
cost: -60
gain 40

sensor 2



coverage: 30
cost: - 1
gain 29

sensor 3



coverage: 30
cost: - 1
gain 29

sensor 4

coverage: 40
cost: - 3
gain 37

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

Greedy can fail ...

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

greedy solution:

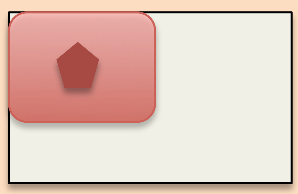
$F(A) = 40$

optimal solution: $F(A) = 95$

sensor 1

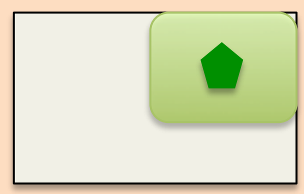
coverage: 100
cost: -60
gain 40

sensor 2



coverage: 30
cost: - 1
gain 29

sensor 3

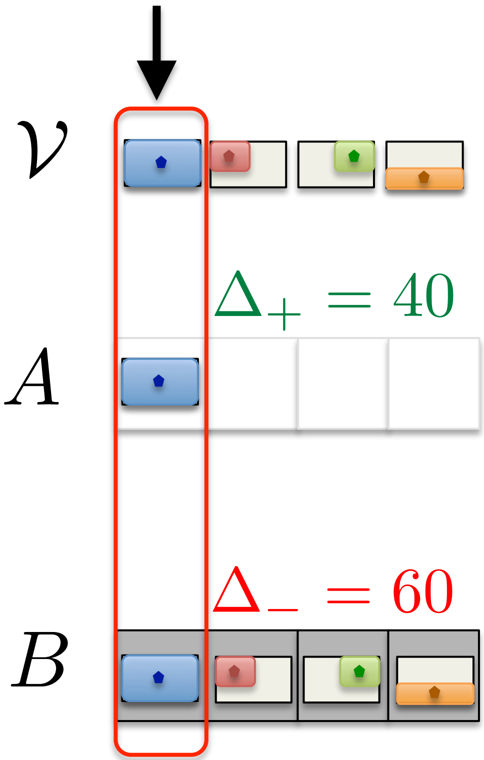


coverage: 30
cost: - 1
gain 29

sensor 4

coverage: 40
cost: - 3
gain 37

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):

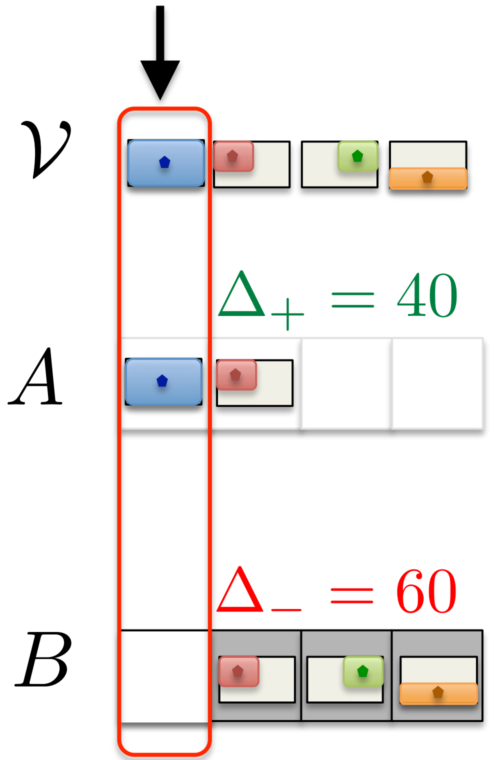
$$\Delta_- = [\, F(B \setminus a) - F(B) \,]_+$$

add with probability

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

coverage: 100
cost: -60

Double (bidirectional) greedy



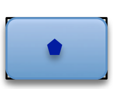
Start: $A = \emptyset,\ B = \mathcal{V}$

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

add with probability

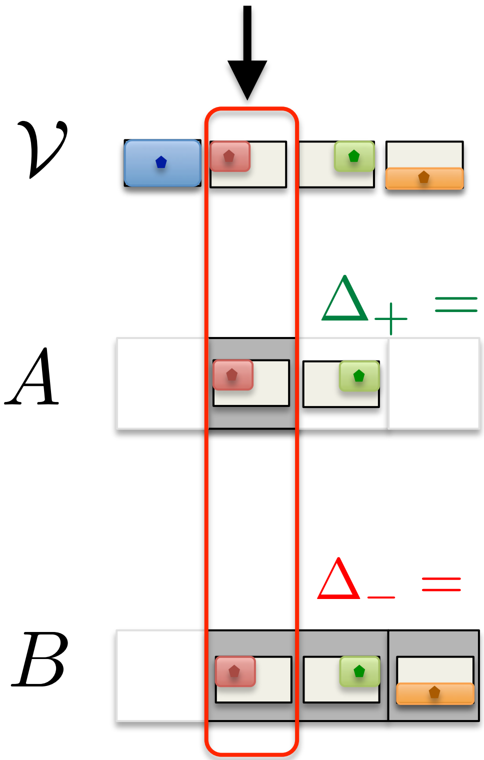
$$\mathbb{P}(\text{add}) = \frac{\Delta_+}{\Delta_+ + \Delta_-}$$

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?*

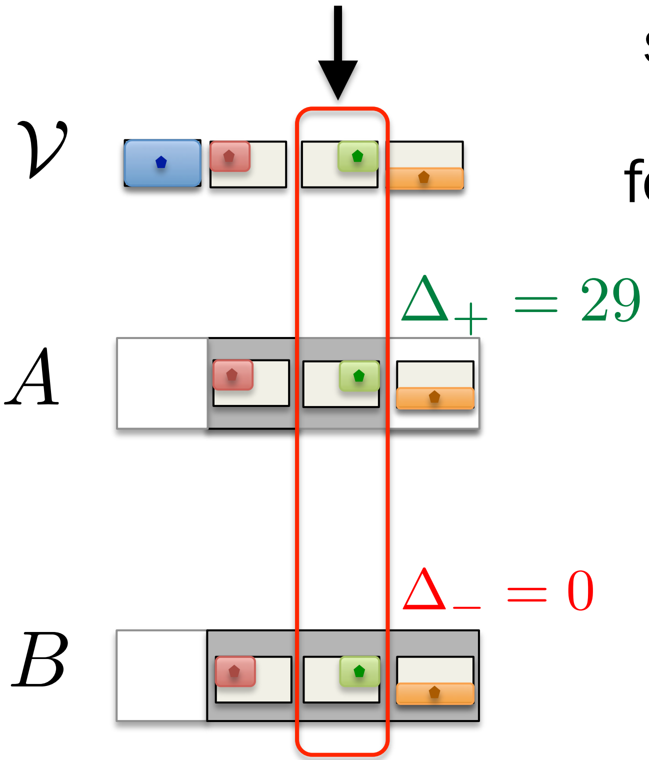
add with probability

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

$\boxed{\text{add to A}}$ 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

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

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

Non-monotone maximization

- alternatives to double greedy?
  local search *(Feige et al 2007)*

- constraints?
  possible, but different algorithms

- distributed algorithms? yes!
  - divide-and-conquer as before *(de Ponte Barbosa et al 2015)*
  - concurrency control / Hogwild *(Pan et al 2014)*

Submodular maximization: summary

- many applications: diverse, informative subsets

- NP-hard, but greedy or local search
- distinguish monotone / non-monotone

- several constraints possible
  (monotone and non-monotone)

Submodularity and machine learning

distributions over labels, sets
log-submodular/
supermodular probability
e.g. "attractive" graphical models,
determinantal point processes

submodularity
& machine
learning!

diffusion processes,
covering, rank,
connectivity,
entropy,
economies of scale,
summarization, …
submodular
phenomena

(convex) regularization
submodularity: "discrete
convexity"
e.g. combinatorial sparse estimation