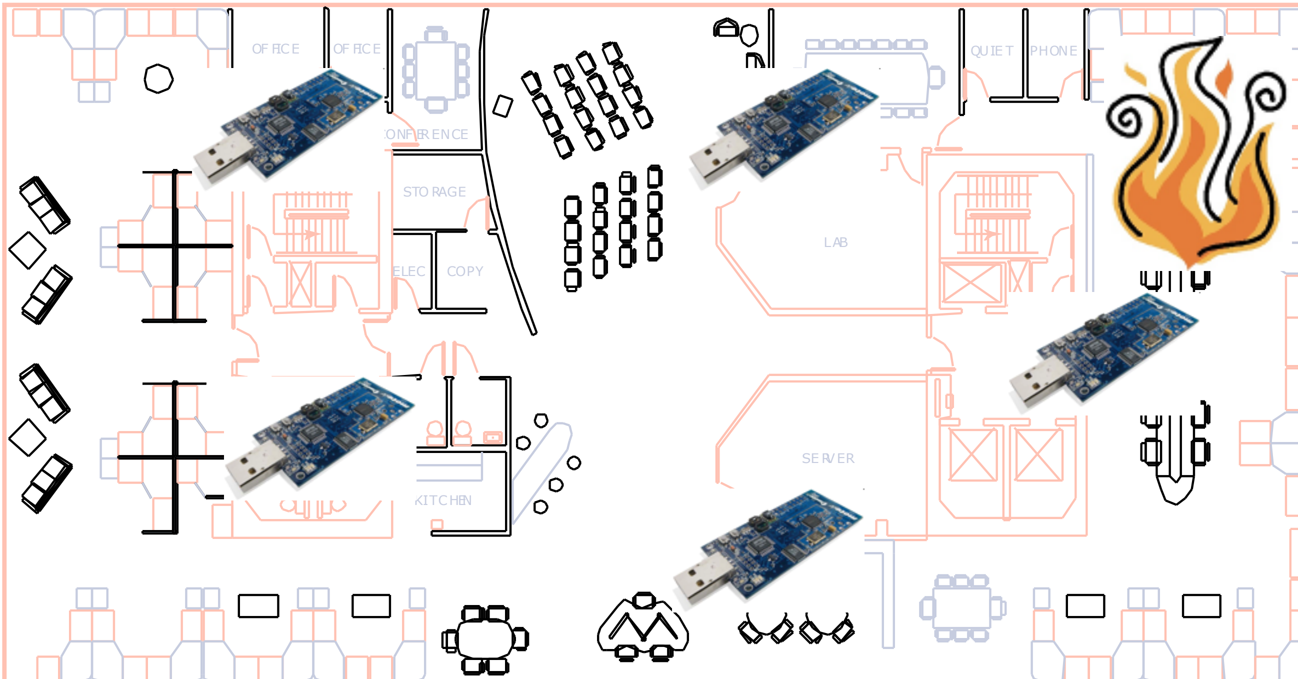
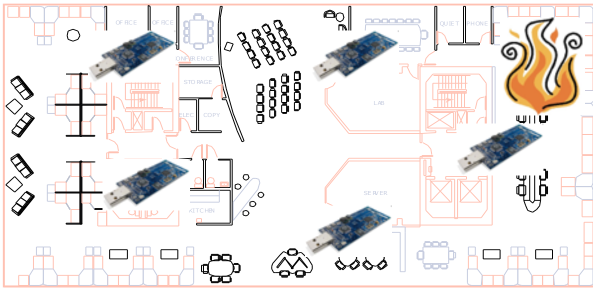
Example: placing sensors



Place sensors to monitor temperature

Set functions

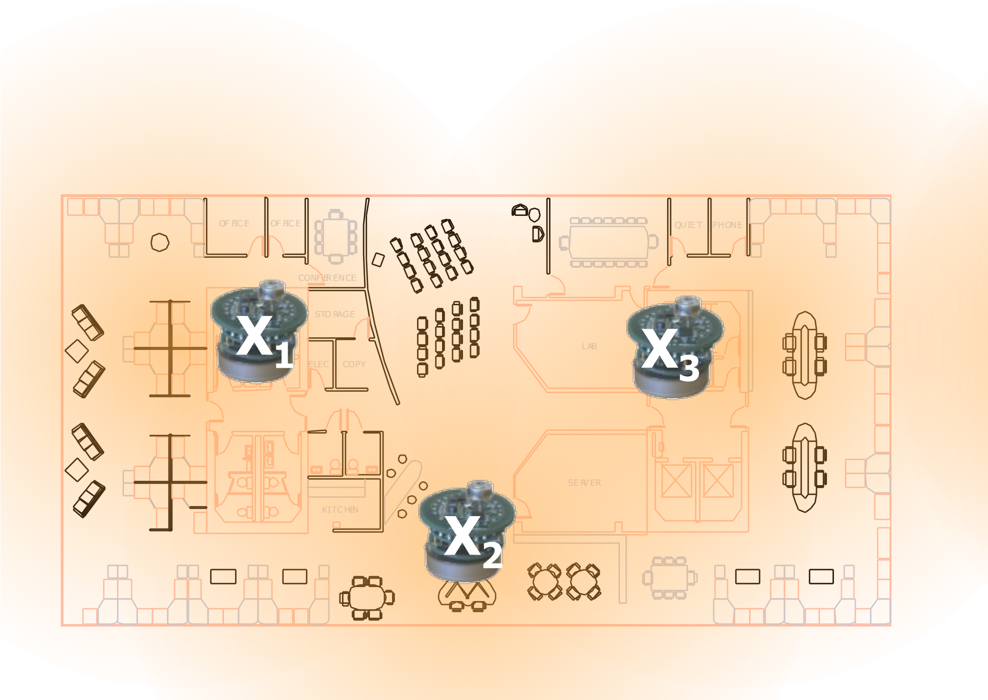
- finite ground set $V = \{1, 2, \ldots, n\}$
- set function $F : 2^V \to \mathbb{R}$



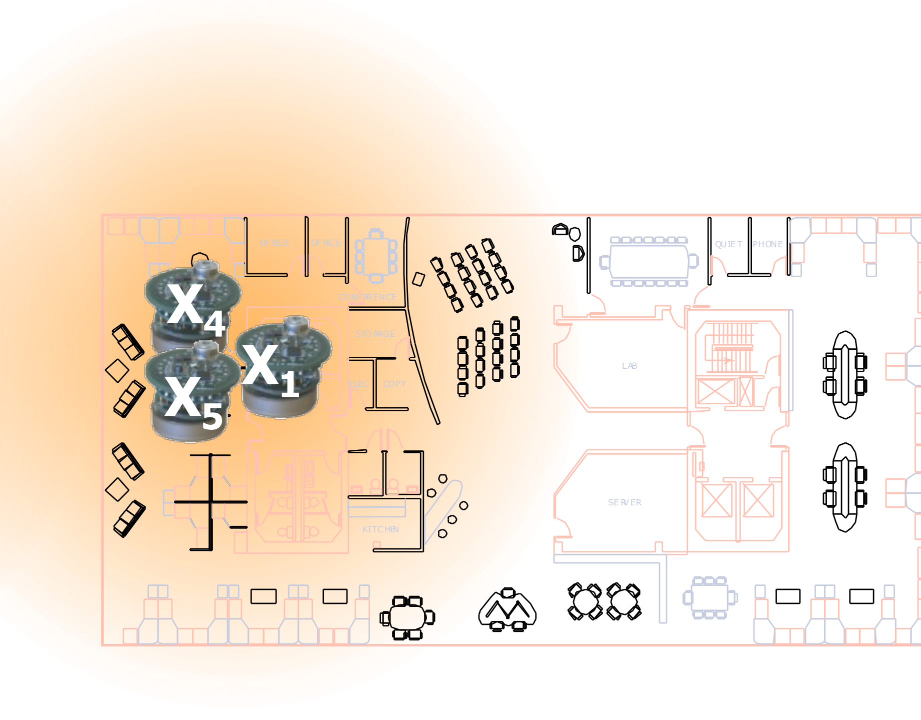
- will assume $F(\emptyset) = 0$ (w.l.o.g.)
- assume black box that can evaluate $F(A)$ for any $A \subseteq V$

Example: placing sensors

Utility $F(A)$ of having sensors at subset A of all locations



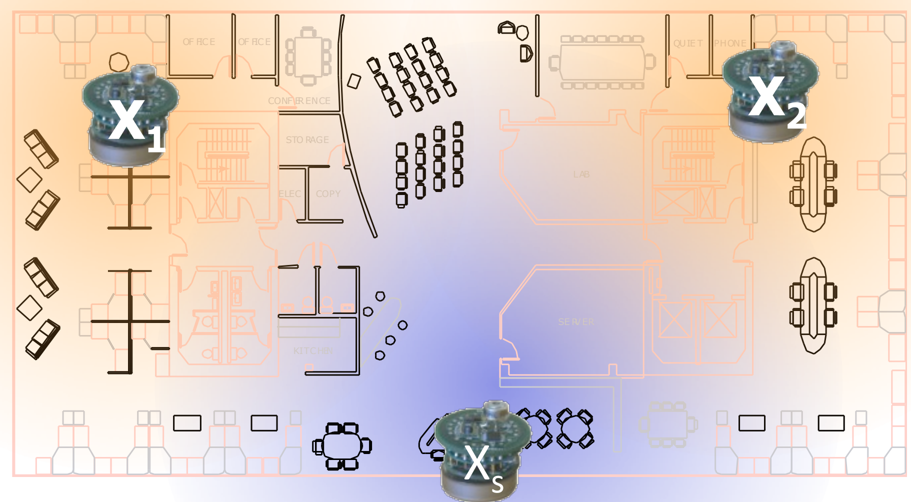
A={1,2,3}: Very informative
High value F(A)



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

Marginal gain

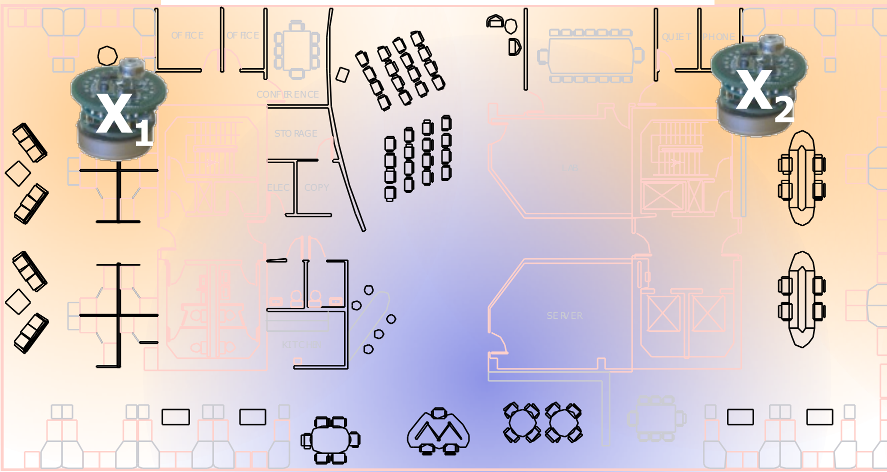
- Given set function $F : 2^V \rightarrow \mathbb{R}$
- Marginal gain: $\Delta_F(s \mid A) = F(\{s\} \cup A) - F(A)$



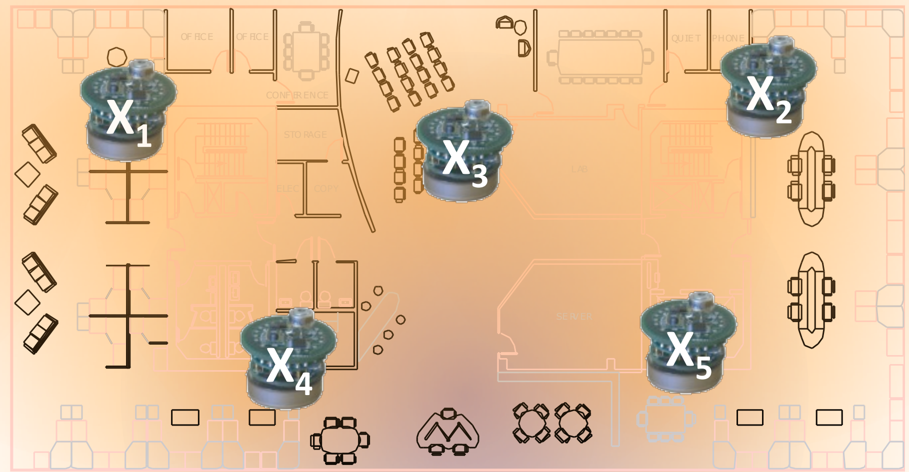
new sensor s

Decreasing gains: submodularity

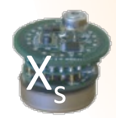
placement A = {1,2}



placement B = {1,…,5}



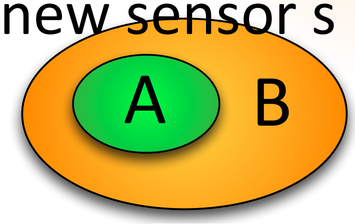
Big gain



new sensor s

small gain

+ • s



+ • s

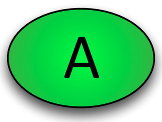
$$A \subseteq B$$

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

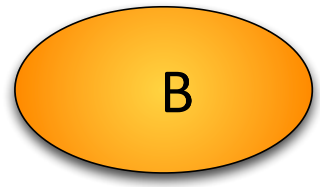
$$\Delta(s \mid A)$$

Equivalent characterizations

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



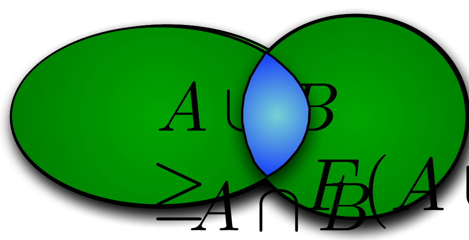
$+ \bullet$ s



$+ \bullet$ s

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

- Union-Intersection: for all $A, B \subseteq V$



$$F(A) + F(B) \quad \geq \quad F(A \cup B) + F(A \cap B)$$

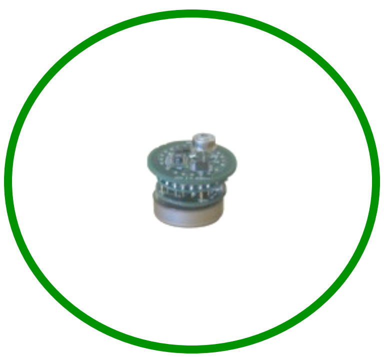
Questions

How do I prove my problem is submodular?

Why is submodularity useful?

Example: Set cover

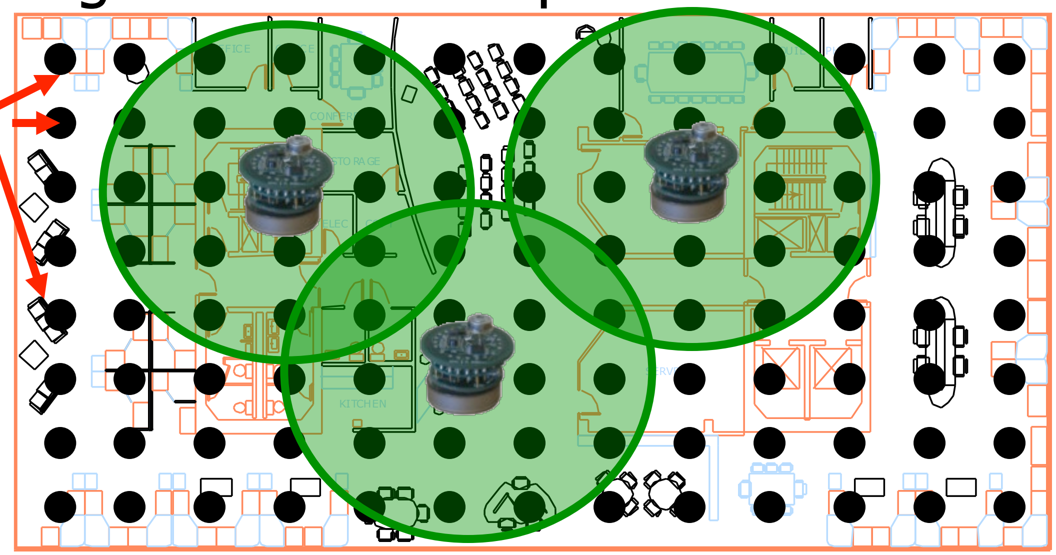
place sensors
in building



Node predicts
values of positions
with some radius

Possible
locations
V

goal: cover floorplan with discs



$A \subseteq V$: $F(A) =$

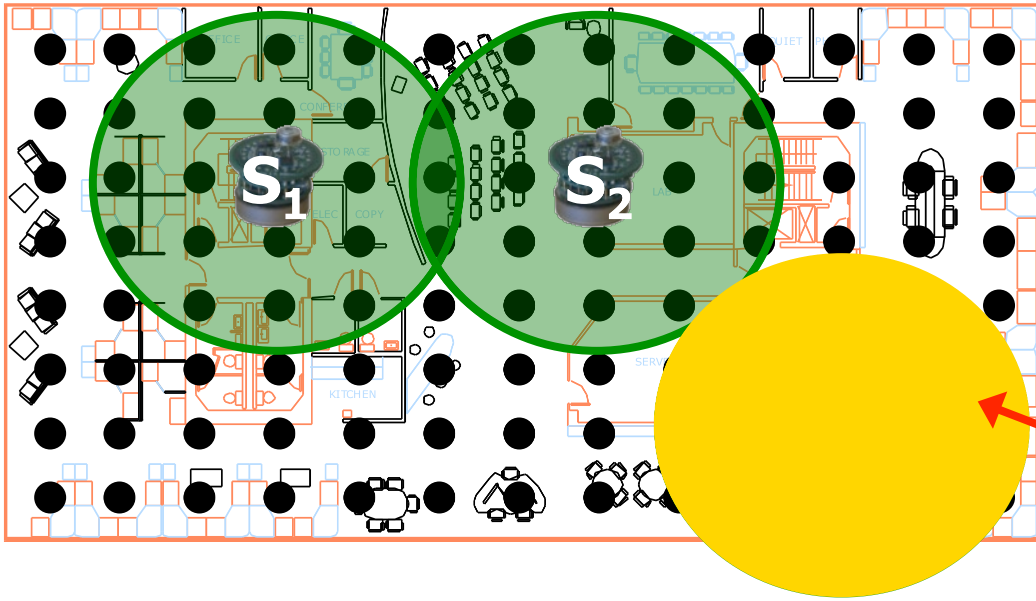
"area covered by sensors placed at A"

Formally:

Finite set W, collection of n subsets $S_i \subseteq W$

For $A \subseteq V$ define $F(A) = \left|\bigcup_{i \in A} S_i\right|$

Set cover is submodular

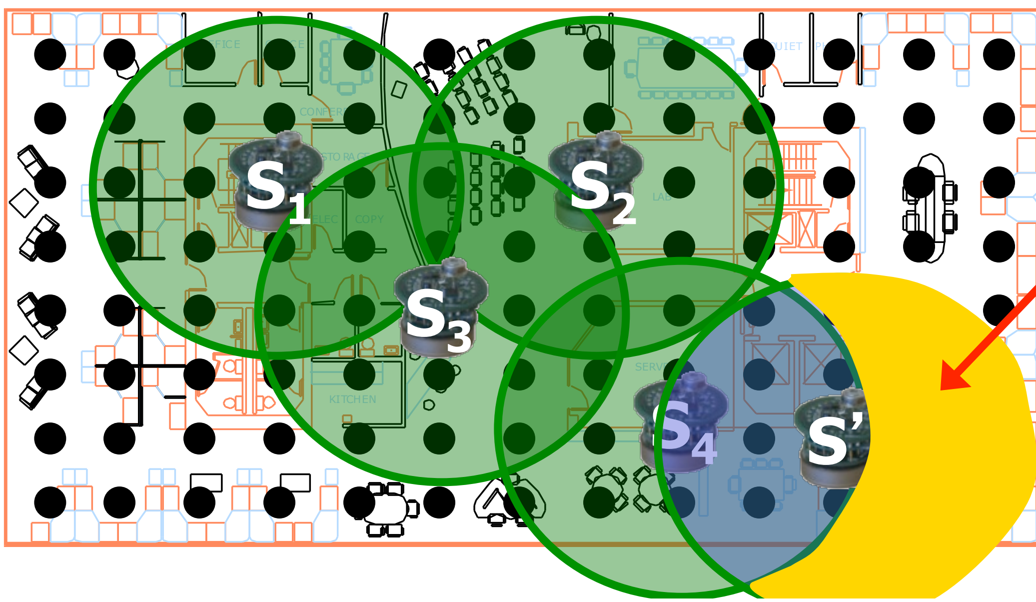


$A=\{s_1,s_2\}$

$F(A \cup \{s'\}) - F(A)$

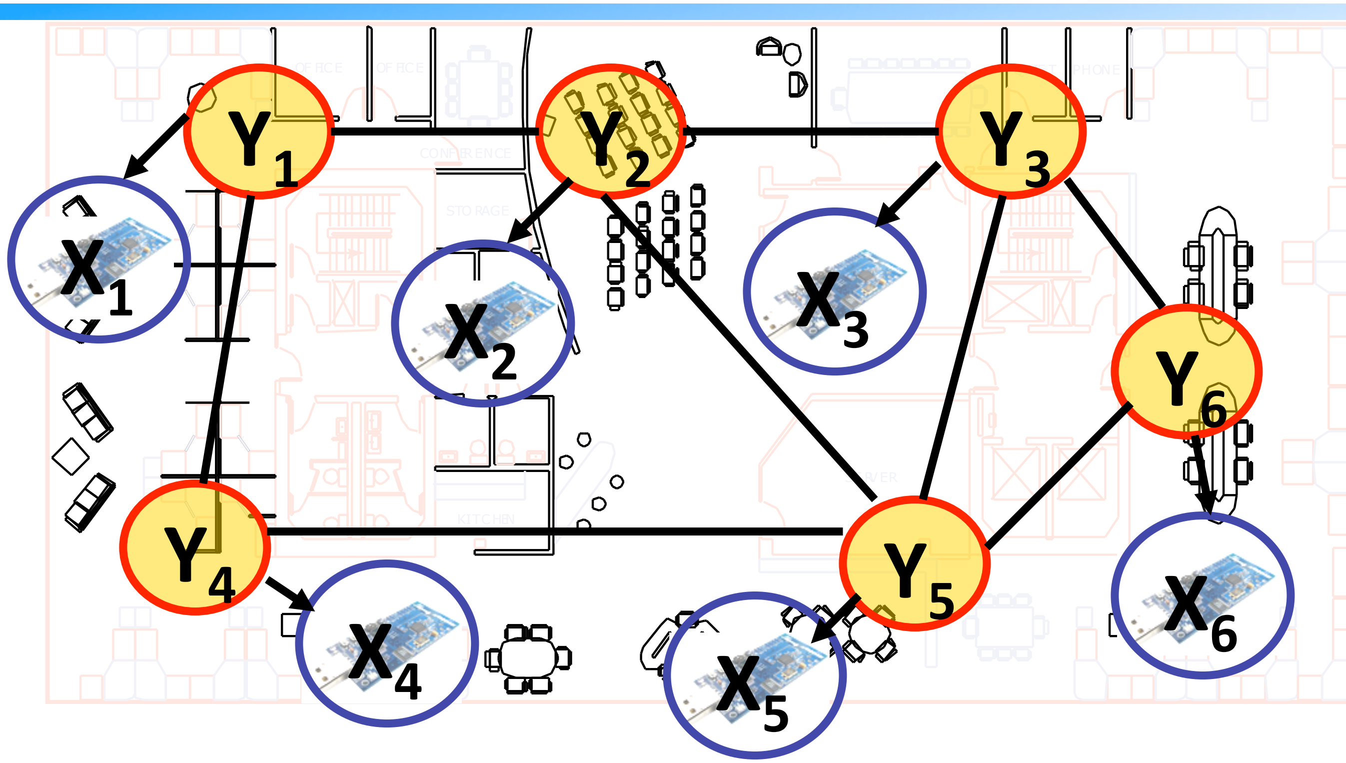
$\geq$

$F(B \cup \{s'\}) - F(B)$



$B = \{s_1,s_2,s_3,s_4\}$

More complex model for sensing



Y_s: temperature at location s

X_s: sensor value at location s

$X_s = Y_s$ + noise

Joint probability distribution

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

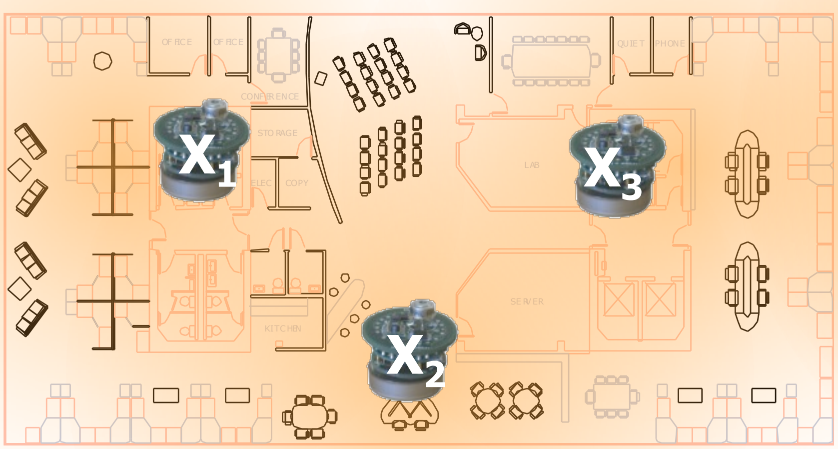
Example: Sensor placement

Utility of having sensors at subset A of all locations

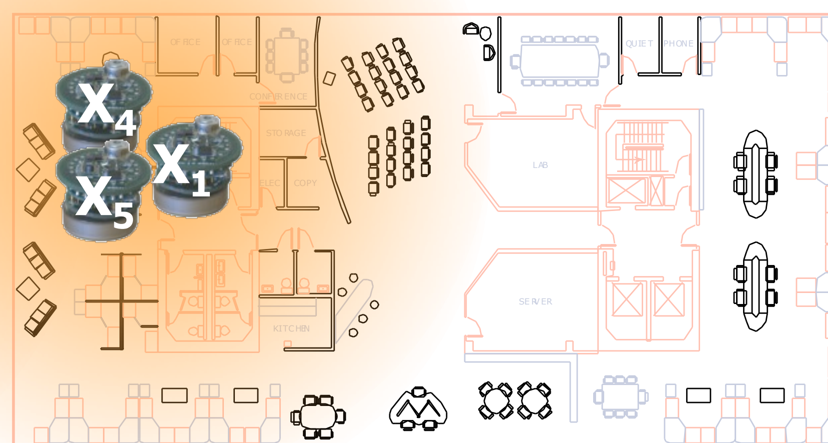
$$F(A) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X}_A)$$

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)

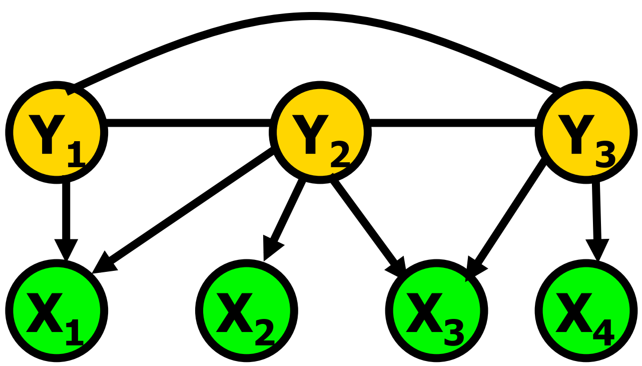
Submodularity of Information Gain

$Y_1,\ldots,Y_m$, X_1, …, X_n discrete RVs

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

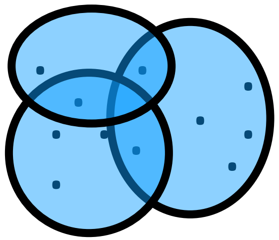
- F(A) is NOT always submodular

If X_i are all conditionally independent given Y, then F(A) is submodular! [Krause & Guestrin `05]

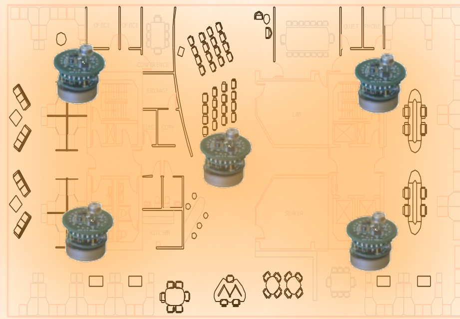


Proof:
"information never hurts"

Submodular maximization



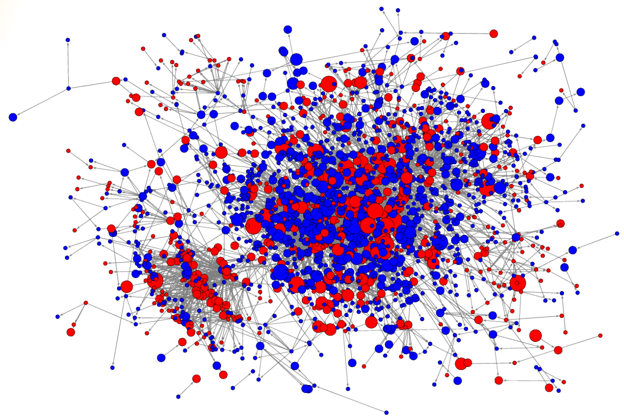
covering



sensing

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

summarization



network inference

Submodular maximization

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

➔ submodularity and concavity

Concave aspects

- submodularity:

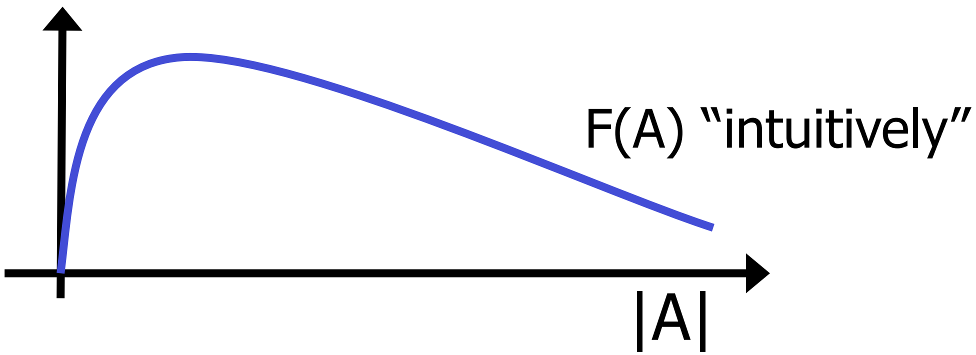
$$A \subseteq B, \quad s \notin B:$$

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

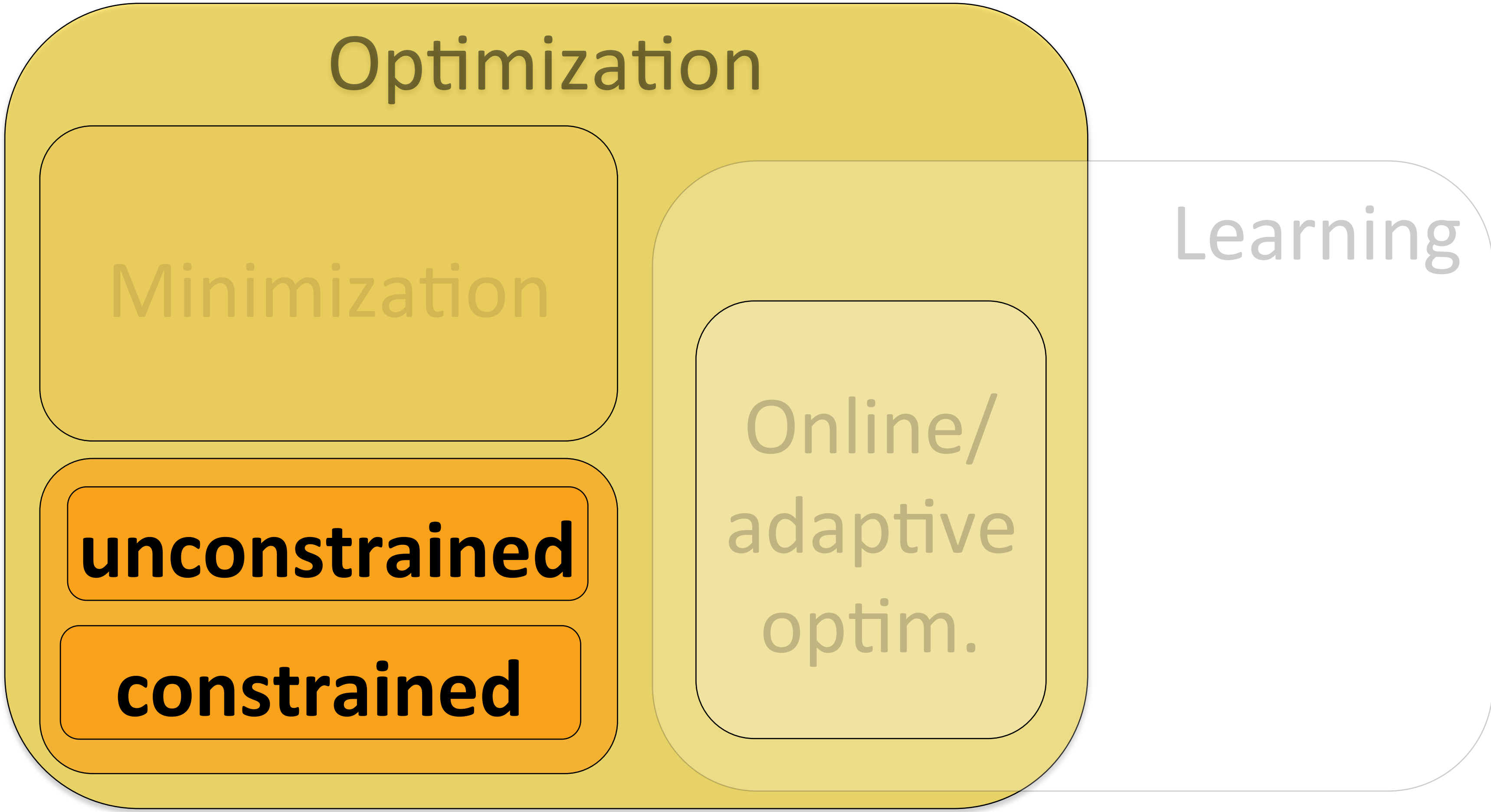
- concavity:

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

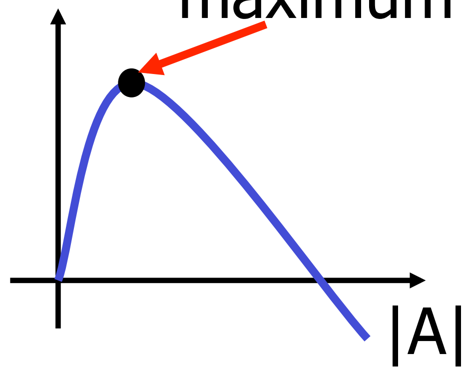
$$f(a+s) - f(a) \quad \geq \quad f(b+s) - f(b)$$



Optimization



Maximizing submodular functions



- Suppose we want for submodular F

$$A^* = \arg\max_A F(A) \text{ s.t. } A \subseteq V$$

- Example:
  - F(A) = U(A) – C(A) where U(A) is submodular utility, and C(A) is supermodular cost function

- In general: NP hard. Moreover:
- If F(A) can take negative values:
  As hard to approximate as maximum independent set (i.e., NP hard to get $O(n^{1-\varepsilon})$ approximation)

Unconstrained vs. constraint maximization

Given monotone utility F(A) and cost C(A), optimize:

<u>Option 1:</u>

$$\boxed{\begin{aligned} &\max_A F(A) - C(A) \\ &\text{s.t. } A \subseteq V \end{aligned}}$$

"Scalarization"

Can get 1/2 approx…

if F(A)-C(A) ≥ 0

for all sets A

<u>Option 2:</u>

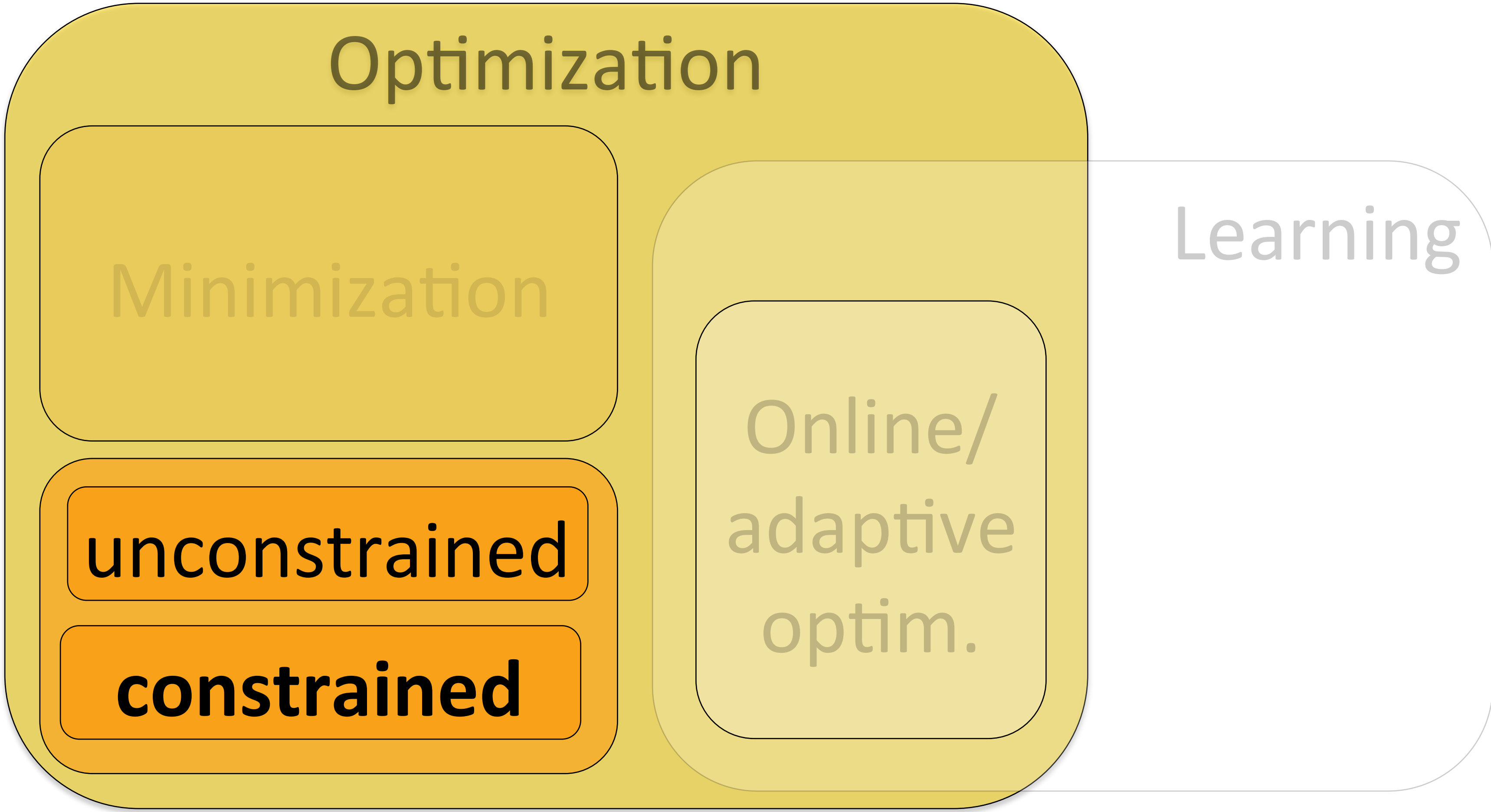
$$\boxed{\begin{aligned} &\max_A F(A) \\ &\text{s.t. } C(A) \leq B \end{aligned}}$$

"Constrained maximization"

What is possible?

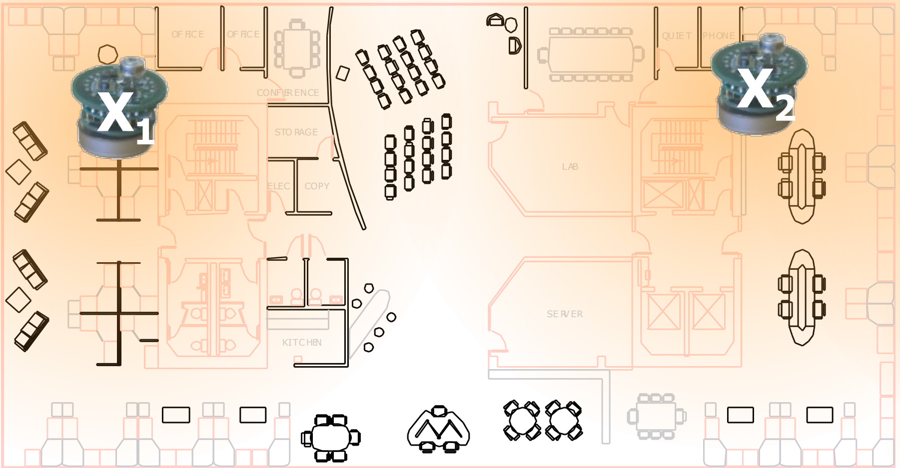
Positiveness is a strong requirement ☹

Optimization

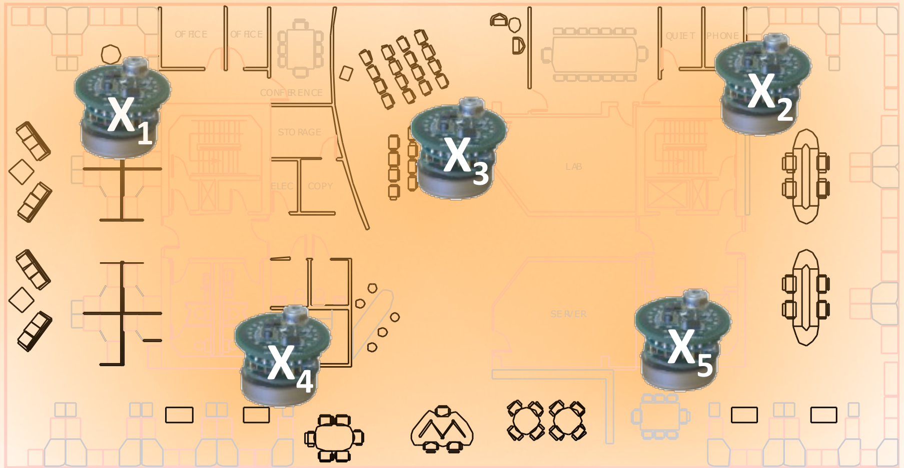


Monotonicity

Placement A = {1,2}



Placement B = {1,…,5}



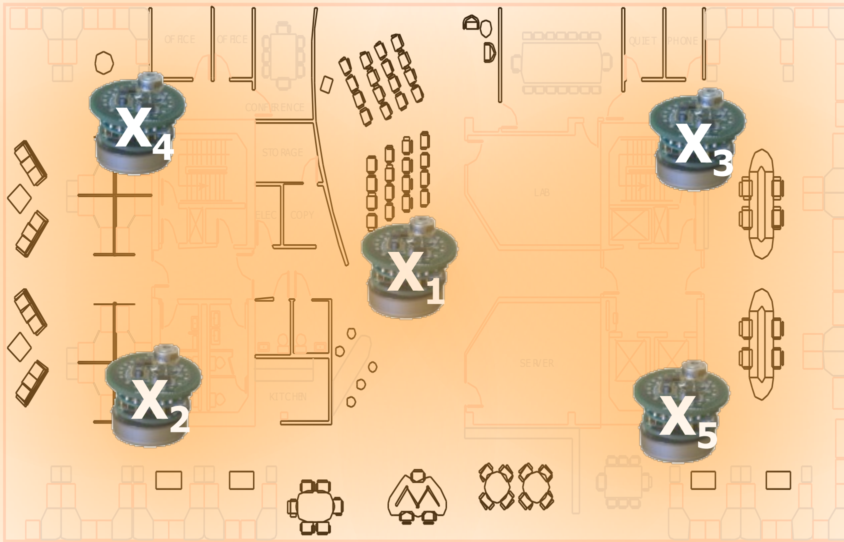
F is monotonic: $$\forall A, s : \underbrace{F(A \cup \{s\}) - F(A)}_{\Delta(s \mid A) \geq 0} \geq 0$$

Adding sensors can only help

Cardinality constrained maximization

- **Given**: finite set V, monotone SF F
- **Want**: $$\boxed{\begin{array}{l}\mathcal{A}^* \subseteq \mathcal{V} \quad \text{such that} \\ \mathcal{A}^* = \underset{|\mathcal{A}| \leq k}{\operatorname{argmax}} F(\mathcal{A})\end{array}}$$

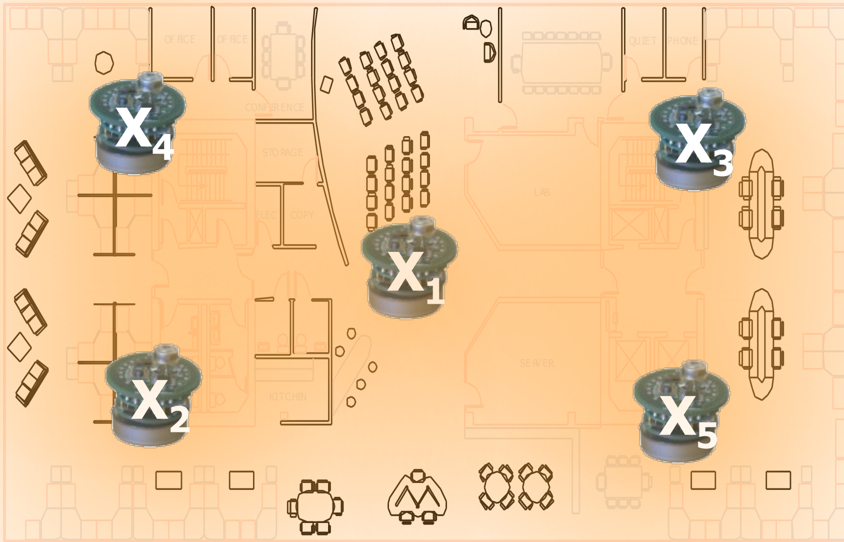
NP-hard!



Greedy algorithm

- **Given**: finite set V, monotone SF F
- **Want**: $\mathcal{A}^* \subseteq \mathcal{V}$ such that
  $$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \le k} F(\mathcal{A})$$

NP-hard!



Greedy algorithm:

Start with $\mathcal{A} = \emptyset$

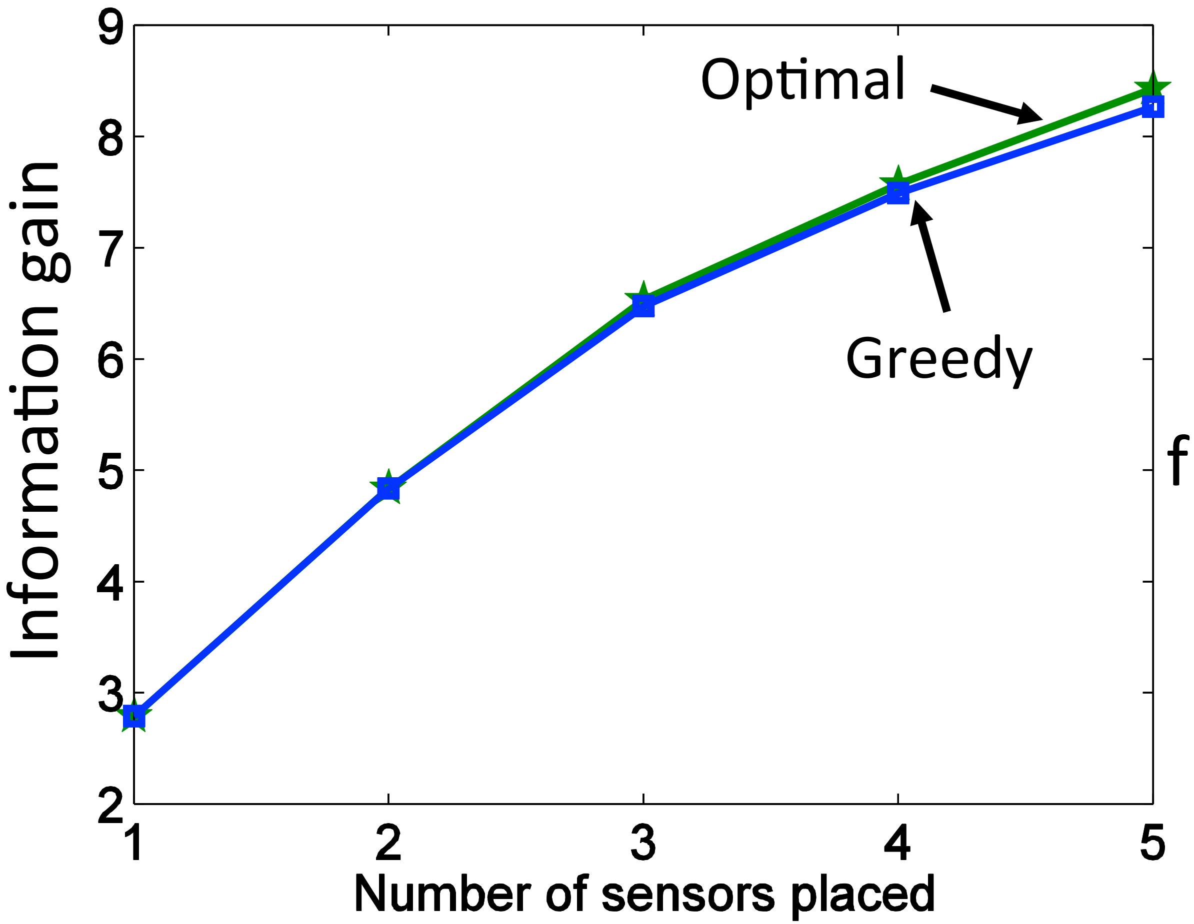
For i = 1 to k

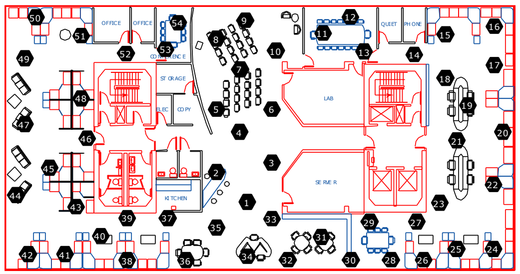
$$s^* \leftarrow \arg\max_s F(\mathcal{A} \cup \{s\})$$

$$\mathcal{A} \leftarrow \mathcal{A} \cup \{s^*\}$$

How well can this simple heuristic do?

Performance of greedy





Temperature data from sensor network

Greedy empirically close to optimal. Why?

One reason submodularity is useful

Theorem [Nemhauser, Fisher & Wolsey '78]

For monotonic submodular functions,
Greedy algorithm gives constant factor approximation

$$F(A_{greedy}) \geq \underbrace{(1-1/e)}_{\sim 63\%} F(A_{opt})$$

- Greedy algorithm gives near-optimal solution!
- In general, need to evaluate exponentially many sets to do better!
  [Nemhauser & Wolsey '78]
- Also many special cases are hard (set cover, mutual information, …)

Scaling up the greedy algorithm [Minoux '78]

In round i+1,

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

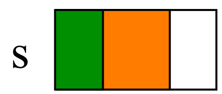
I.e., maximize "marginal benefit" $\Delta(s \mid A_i)$

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

Key observation: Submodularity implies

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

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

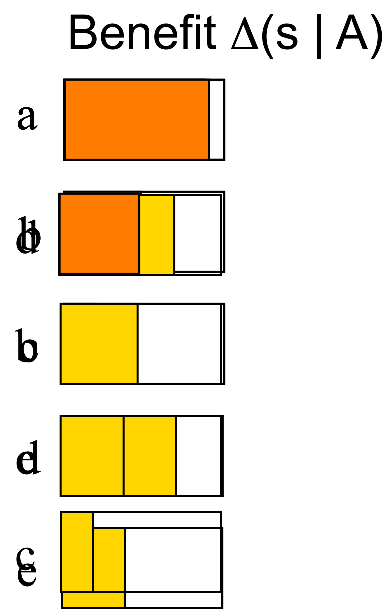


Marginal benefits can never increase!

"Lazy" greedy algorithm [Minoux '78]

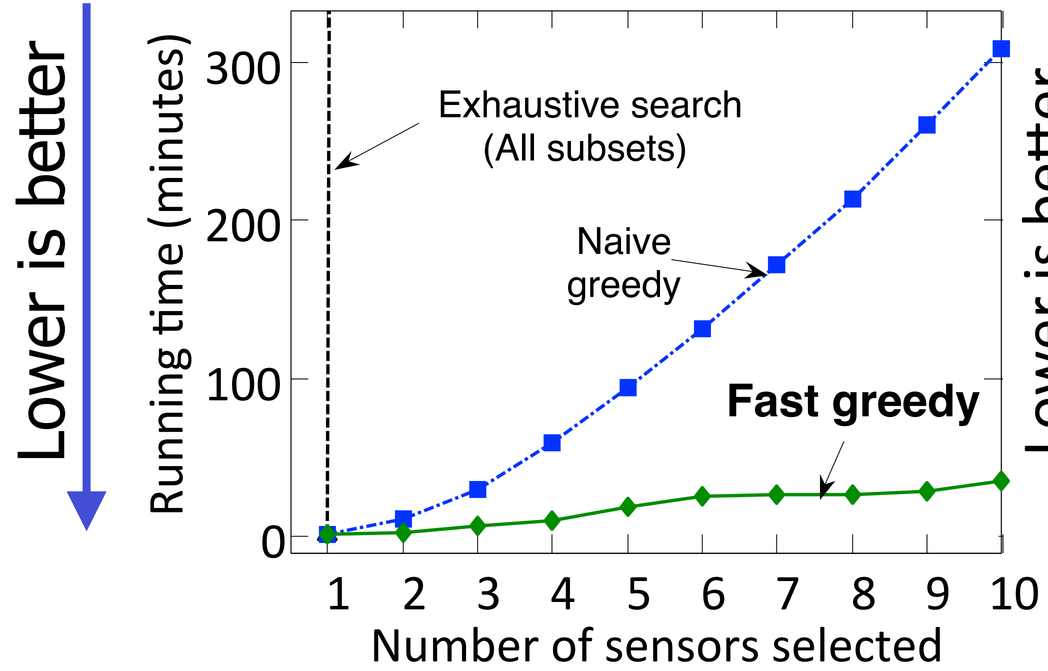
Lazy greedy algorithm:

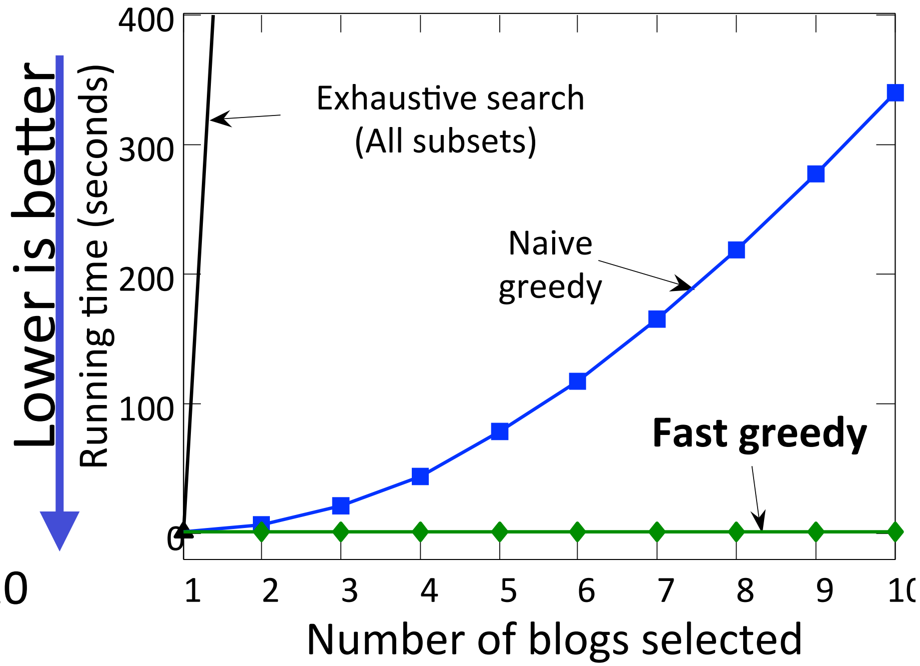
- First iteration as usual
- Keep an ordered list of marginal benefits Δ_i from previous iteration
- Re-evaluate Δ_i only for top element
- If Δ_i stays on top, use it, otherwise re-sort



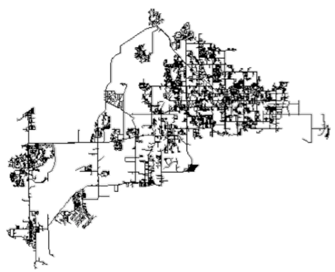
Note: Very easy to compute online bounds, lazy evaluations, etc. [Leskovec, Krause et al. '07]

Empirical improvements [Leskovec, Krause et al'06]





Sensor placement



Blog selection

30x speedup

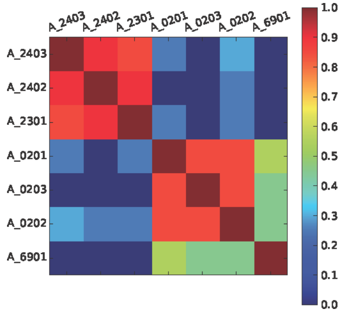
700x speedup

Submodular Sensing Problems

[with Guestrin, Leskovec, Singh, Sukhatme, …]

Environmental monitoring
[UAI'05, JAIR '08, ICRA '10]

Water distribution networks
[J WRPM '08]



Experiment design
[NIPS '10, '11, PNAS'13]

Recommending blogs & news
[KDD '07, '10]

Can all be reduced to monotonic submodular maximization

Maximization: More complex constraints

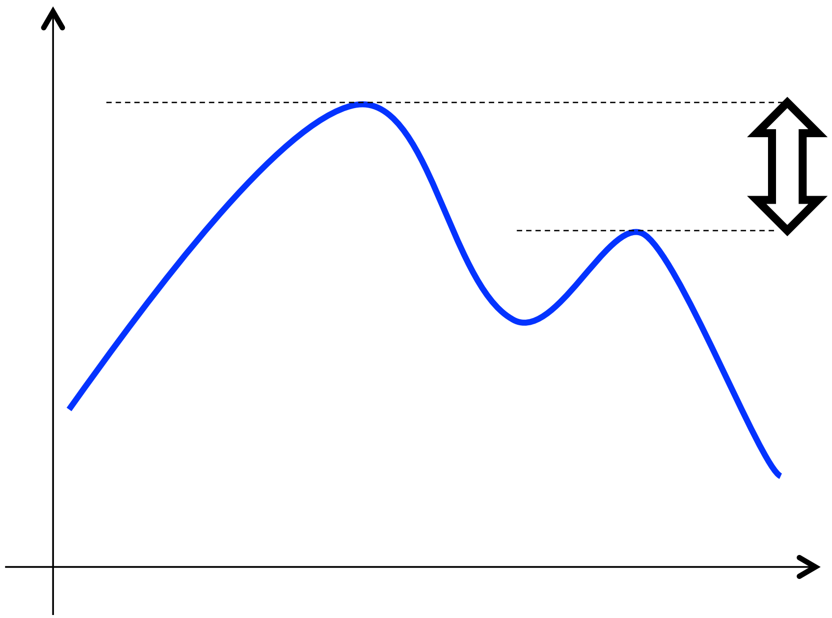
- Approximate submodular maximization possible under a variety of constraints:
  - (Multiple) matroid constraints
  - Knapsack (non-constant cost functions)
  - Multiple matroid and knapsack constraints
  - Path constraints (Submodular orienteering)
  - Connectedness (Submodular Steiner)
  - Robustness (minimax)
  - ...

Greedy works well

Need non-greedy algorithms

- Survey on „Submodular Function Maximization“ [Krause & Golovin ‘12] on submodularity.org

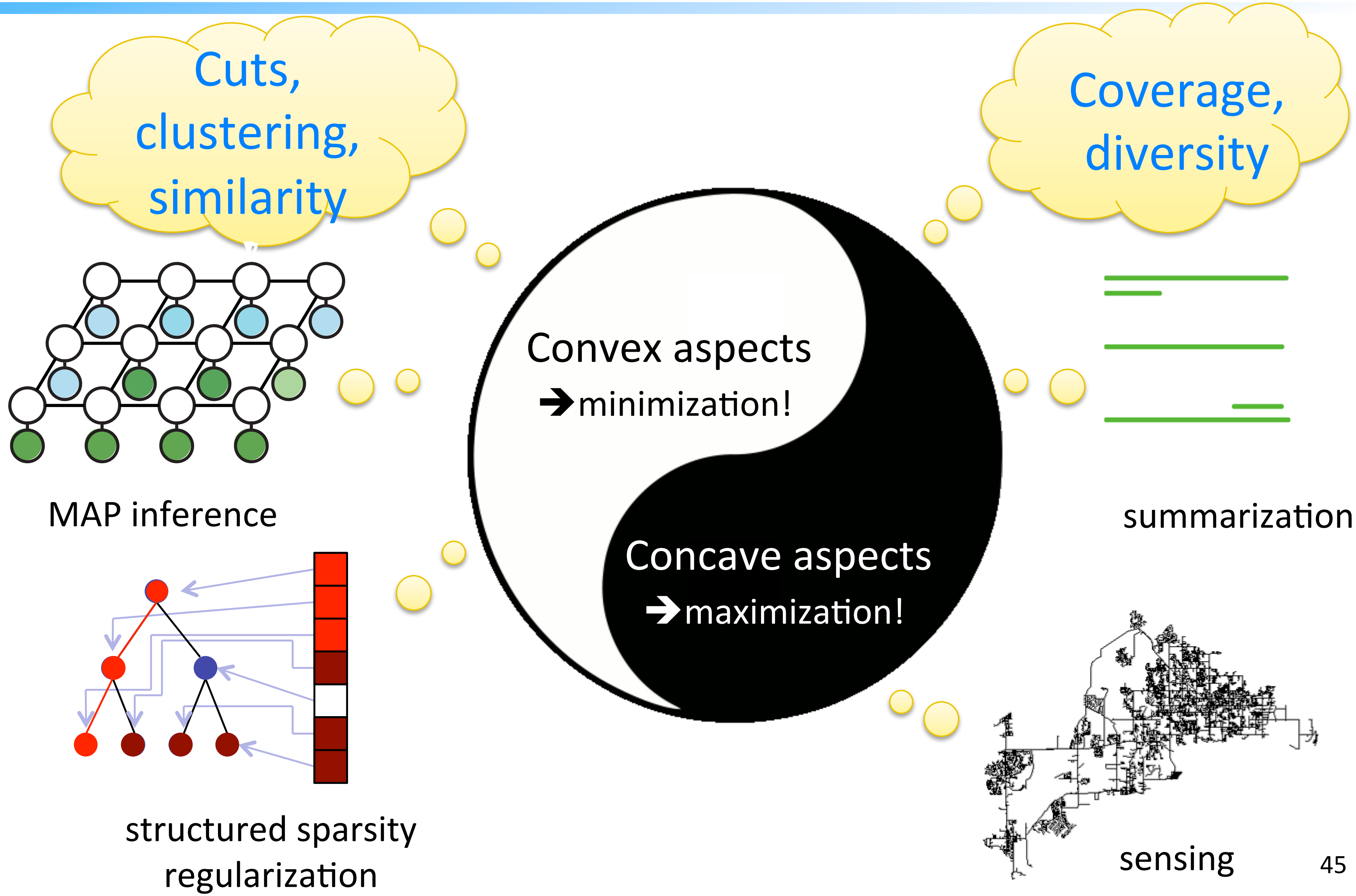
Key intuition for approx. maximization



For submod. functions,
local maxima
can't be too bad

- E.g., all local maxima under cardinality constraints are within factor 2 of global maximum
- Key insight for more complex maximization
  ➔ Greedy, local search, simulated annealing
  for (non-monotone, constrained, ...)

Two-faces of submodular functions



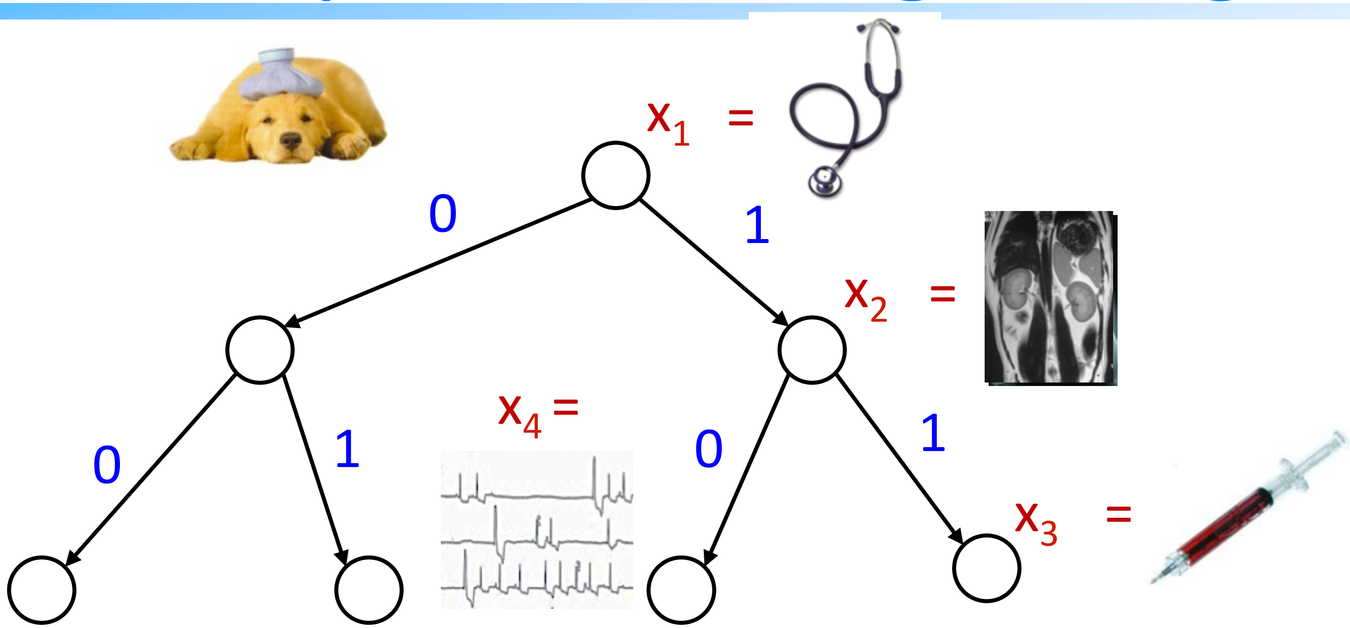
	Maximization	Minimization
Unconstrained	NP-hard, but well-approximable (if nonnegative)	Polynomial time! Generally inefficent (n^6), but can exploit special cases (cuts; symmetry; decomposable; ...)
Constrained	NP-hard but well-approximable „Greedy-(like)“ for cardinality, matroid constraints; Non-greedy for more complex (e.g., connectivity) constraints	NP-hard; hard to approximate, still useful algorithms

Learning to optimize submodular functions

- Online submodular optimization
  - Learn to pick a sequence of sets to maximize a sequence of (unknown) submodular functions
  - *Application*: Making diverse recommendations

- Adaptive submodular optimization
  - Gradually build up a set, taking into account feedback
  - *Application*: Experimental design / Active learning

Adaptive Sensing / Diagnosis



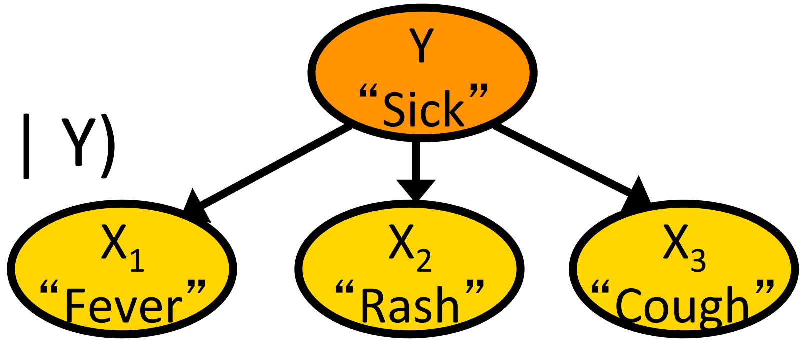
Want to effectively diagnose while minimizing cost of testing!

Classical submodularity does not apply ☹

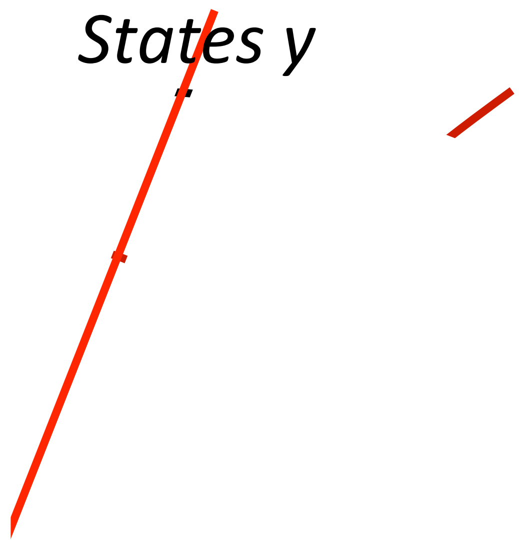
Can we generalize submodularity for sequential decision making?

Adaptive selection in diagnosis

- Prior over diseases P(Y)
- Deterministic test outcomes P($\mathbf{X}_V$ | Y)



- Each test eliminates hypotheses y



Problem Statement

Given:

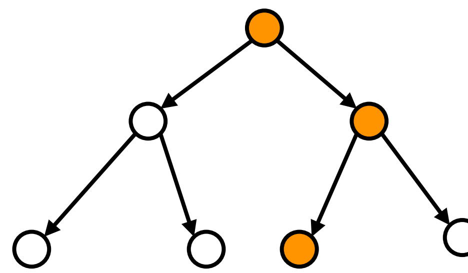
- Items (tests, experiments, actions, …) V={1,…,n}
- Associated with random variables $X_1,\ldots,X_n$ taking values in O
- Objective: $f : 2^V \times O^V \to \mathbb{R}$
- Policy π maps observation $\mathbf{x}_A$ to next item

Value of policy π: $$F(\pi) = \sum_{\mathbf{x}_V} P(\mathbf{x}_V) f(\pi(\mathbf{x}_V), \mathbf{x}_V)$$

Tests run by π
if world in state $\mathbf{x}_V$

Want $$\pi^* \in \operatorname*{argmax}_{|\pi| \le k} F(\pi)$$

NP-hard (also hard to approximate!)



Adaptive greedy algorithm

- Suppose we've seen $X_A = x_{A.}$
- Conditional expected benefit of adding item s:

$$\Delta(s \mid \mathbf{x}_A) = \mathbb{E}\Big[\underbrace{f(A \cup \{s\}, \mathbf{x}_V) - f(A, \mathbf{x}_V)}_{\text{Benefit if world in state } \mathbf{x}_V} \mid \mathbf{x}_A\Big]$$

Conditional on observations $\mathbf{x}_A$

Adaptive Greedy algorithm:

Start with $A = \emptyset$

For i = 1:k

- Pick $s_k \in \operatorname{argmax}_s \Delta(s \mid \mathbf{x}_A)$
- Observe $X_{s_k} = x_{s_k}$
- Set $A \leftarrow A \cup \{s_k\}$

When does this adaptive greedy algorithm work??

Adaptive submodularity
[Golovin & Krause, JAIR 2011]

Adaptive monotonicity:

$$\Delta(s \mid \mathbf{x}_A) \geq 0$$

x_B observes more than x_A

Adaptive submodularity:

$$\Delta(s \mid \mathbf{x}_A) \geq \Delta(s \mid \mathbf{x}_B) \quad \text{whenever } \mathbf{x}_A \preceq \mathbf{x}_B$$

Theorem: If f is adaptive submodular and adaptive monotone w.r.t. to distribution P, then

$F(\pi_{greedy}) \geq (1-1/e)\, F(\pi_{opt})$

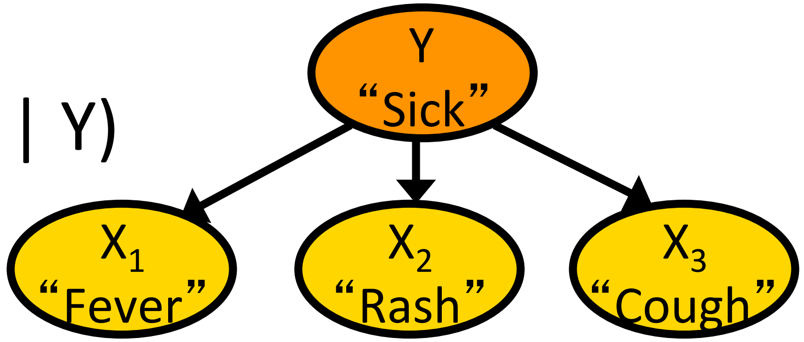
Many other results about submodular set functions can also be "lifted" to the adaptive setting!

From sets to policies

Submodularity	**Adaptive submodularity**
Applies to: set functions	policies, value functions
$\Delta_F(s \mid A) = F(A \cup \{s\}) - F(A)$	$\Delta_F(s \mid \mathbf{x}_A) = \mathbb{E}\left[f(A \cup \{s\}, \mathbf{x}_V) - f(A, \mathbf{x}_V) \mid \mathbf{x}_A\right]$
$\Delta_F(s \mid A) \geq 0$	$\Delta_F(s \mid \mathbf{x}_A) \geq 0$
$A \subseteq B \Rightarrow \Delta_F(s \mid A) \geq \Delta_F(s \mid B)$	$\mathbf{x}_A \preceq \mathbf{x}_B \Rightarrow \Delta_F(s \mid \mathbf{x}_A) \geq \Delta_F(s \mid \mathbf{x}_B)$
$\max_A F(A)$	$\max_\pi F(\pi)$
Greedy algorithm provides	Greedy policy provides
• *(1-1/e)* for max. w card. const.	• *(1-1/e)* for max. w card. const.
• *1/(p+1)* for p-indep. systems	• *1/(p+1)* for p-indep. systems
• *log Q* for min-cost-cover	• *log Q* for min-cost-cover
• *4* for min-sum-cover	• *4* for min-sum-cover

Optimal Diagnosis

- Prior over diseases P(Y)
- Deterministic test outcomes P($\mathbf{X}_V$ | Y)

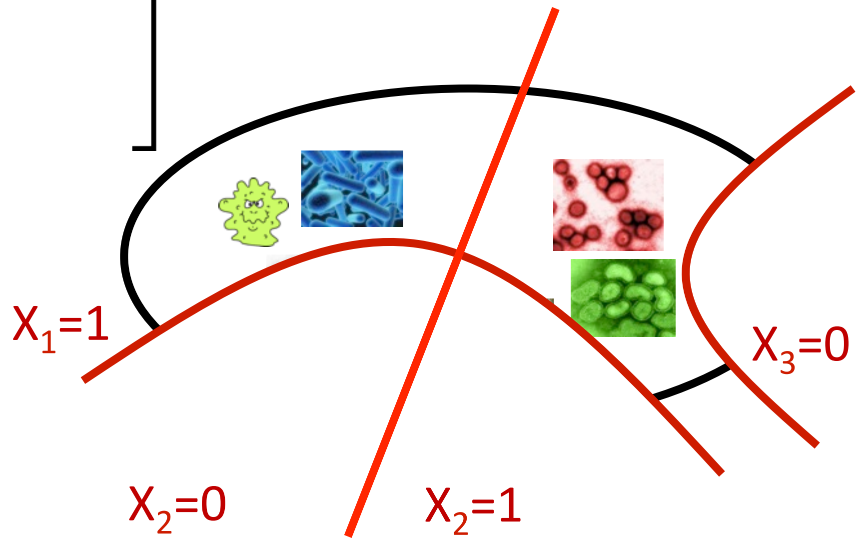


- How should we test to eliminate all incorrect hypotheses?

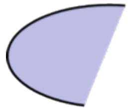
$$\Delta(t \mid x_A) = \mathbb{E}\left[\begin{array}{l}\text{mass ruled out}\\ \text{by } t \text{ if we}\\ \text{know } x_A\end{array}\right]$$

"Generalized binary search"

Equivalent to max. infogain



OD is Adaptive Submodular

$b_0 := \mathbb{P}($  $)$

$g_0 := \mathbb{P}($  $)$

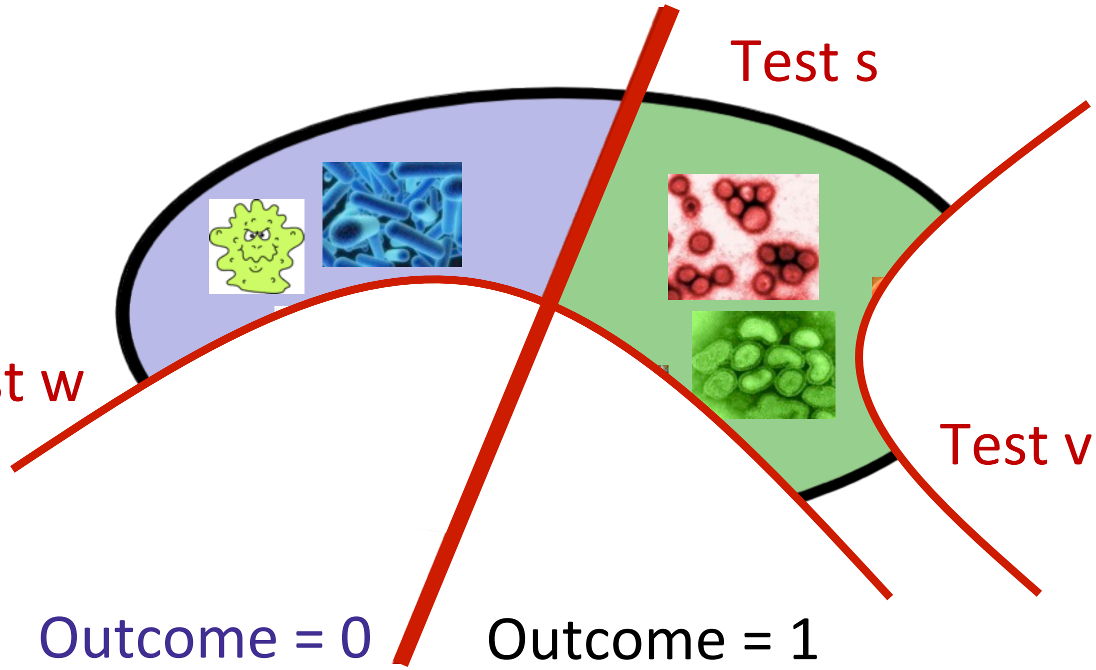
$$\Delta(s \mid \{\}) = \frac{2 g_0 b_0}{g_0 + b_0}$$

$b_1 := \mathbb{P}($  $)$

$g_1 := \mathbb{P}($  $)$

$$\Delta(s \mid \mathbf{x}_{v,w}) = \frac{2 g_1 b_1}{g_1 + b_1}$$

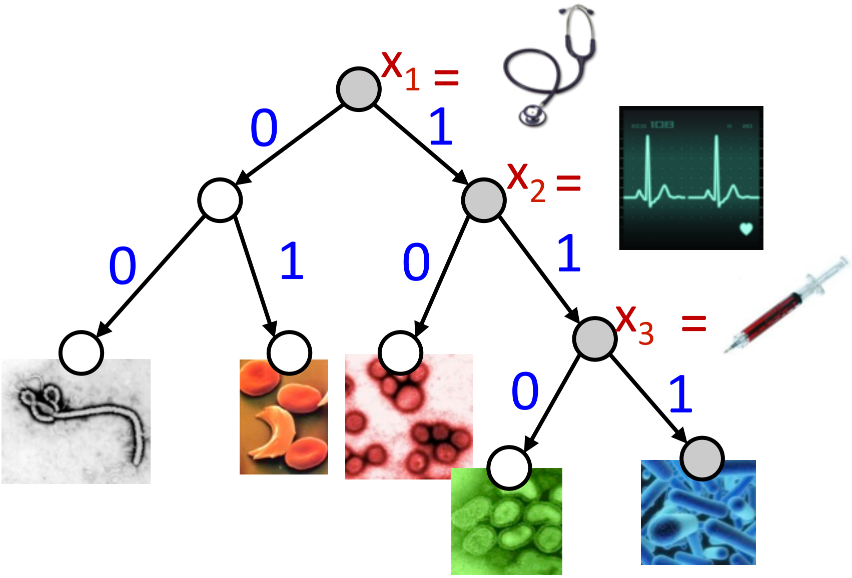
Objective = probability mass of hypotheses you have ruled out.



$b_0 \geq b_1, \quad g_0 \geq g_1$

Not hard to show that $\Delta(s \mid \{\}) \geq \Delta(s \mid \mathbf{x}_{v,w})$

Theoretical guarantees



Garey & Graham, 1974;
Loveland, 1985;
Arkin et al., 1993;
Kosaraju et al., 1999;
Dasgupta, 2004;
Guillory & Bilmes, 2009;
Nowak, 2009;
Gupta et al., 2010

Adaptive-Greedy is a $\boxed{(\ln(1/p_{\min}) + 1)}$ approximation.

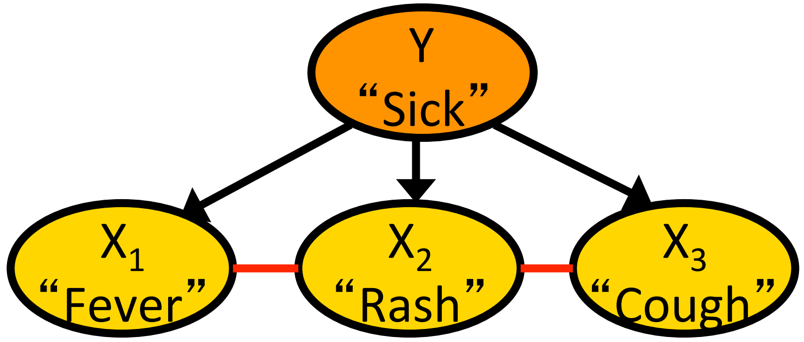
With adaptive submodular analysis!

Result requires that tests are *exact* (no noise)!

What if there is noise?

[w Daniel Golovin, Deb Ray, NIPS '10]

- Prior over diseases P(Y)
- Noisy test outcomes $P(\mathbf{X}_V \mid Y)$
- How should we test
  to learn about y (infer MAP)?



- Existing approaches:
  - Generalized binary search?
  - Maximize information gain?
  - Maximize value of information?

Not adaptive submodular!

> **Theorem**: All these approaches can have cost **more than n/log n** times the optimal cost!

➔ Is there an adaptive submodular criterion??

Theoretical guarantees
[with Daniel Golovin, Deb Ray, NIPS '10]

Theorem: Equivalence class edge-cutting (EC^2) is adaptive monotone and adaptive submodular.

Suppose $P(\mathbf{x}_V, h) \in \{0\} \cup [\delta, 1]$ for all $\mathbf{x}_V, h$

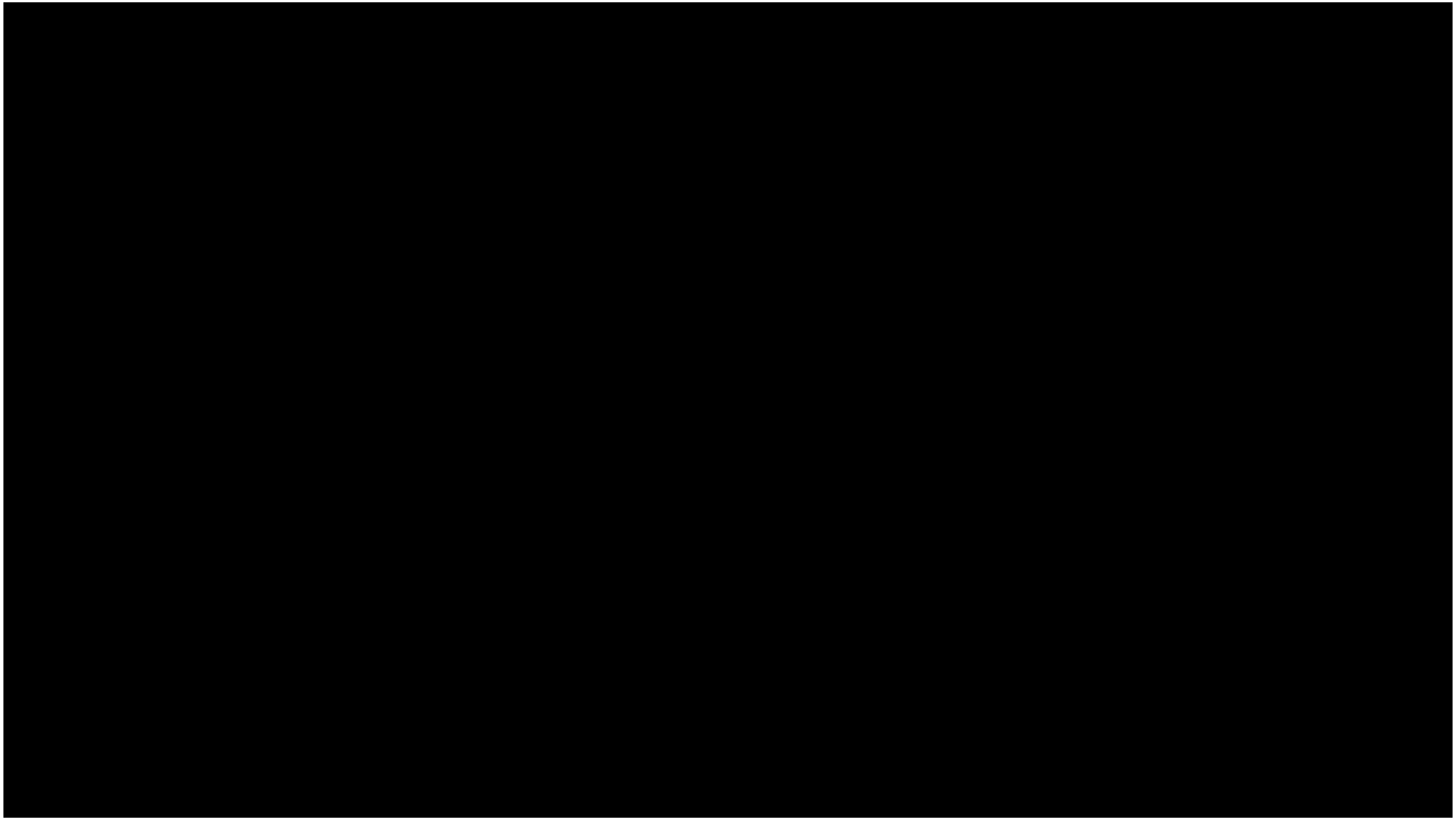
Then it holds that

$$\mathrm{Cost}(\pi_{\mathrm{Greedy}}) \leq \mathcal{O}\left(\log \frac{1}{\delta}\right) \mathrm{Cost}(\pi^*)$$

First approximation guarantees for **nonmyopic VOI** in general graphical models!

Application: Touch-based localization

[Javdani, Klingensmith, Bagnell, Pollard, Srinivasa, ICRA 2013]



Interactive submodular coverage

- Alternative formalization of adaptive optimization [Guillory & Bilmes, ICML '10]
  - Addresses the worst case setting
- Applications to (noisy) active learning, viral marketing [Guillory & Bilmes, ICML '11]