Active Learning and Optimized Information Gathering

Lecture 13 – Submodularity (cont'd)

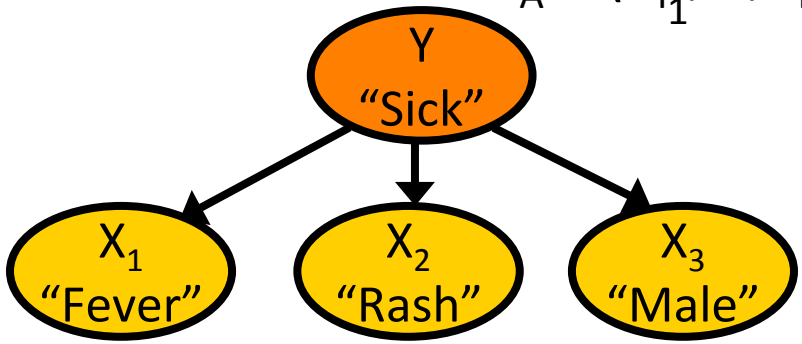
CS 101.2
Andreas Krause

Announcements

- **Homework 2: Due Thursday Feb 19**
- **Project milestone due: Feb 24**
  - 4 Pages, NIPS format: http://nips.cc/PaperInformation/StyleFiles
  - Should contain preliminary results (model, experiments, proofs, …) as well as timeline for remaining work
  - Come to office hours to discuss projects!
- Office hours
  - Come to office hours before your presentation!
  - Andreas: **Monday 3pm-4:30pm**, 260 Jorgensen
  - Ryan: Wednesday 4:00-6:00pm, 109 Moore

Feature selection

- Given random variables $Y, X_1, \ldots X_n$
- Want to predict Y from subset $X_A = (X_{i_1},\ldots,X_{i_k})$



Naïve Bayes Model

Want k most informative features:

$$A^* = \text{argmax } IG(X_A; Y) \text{ s.t. } |A| \leq k$$

where $IG(X_A; Y) = H(Y) - H(Y \mid X_A)$

Uncertainty before knowing X_A

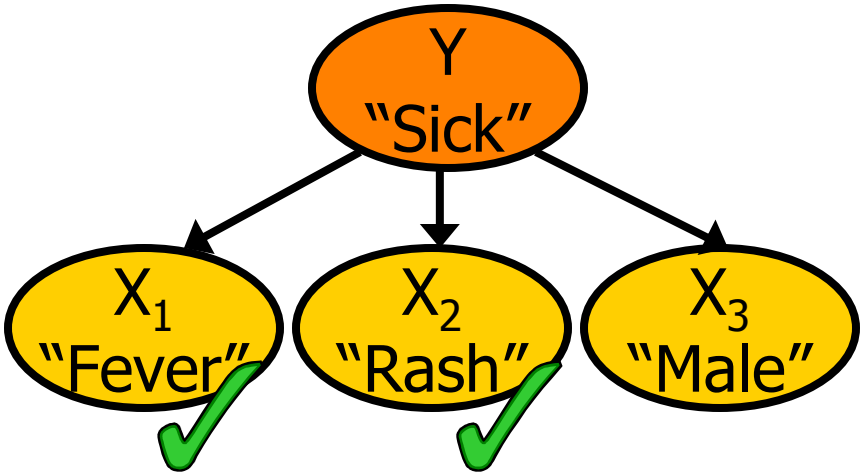
Uncertainty after knowing X_A

Example: Greedy algorithm for feature selection

- Given: finite set V of features, utility function $F(A) = IG(X_A; Y)$
- Want: $A^* \subseteq V$ such that

$$\mathcal{A}^* = \underset{|\mathcal{A}| \leq k}{\operatorname{argmax}} F(\mathcal{A})$$

NP-hard!



<u>Greedy algorithm:</u>

Start with $A = \emptyset$

For i = 1 to k

$s^* := \operatorname{argmax}_s F(A \cup \{s\})$

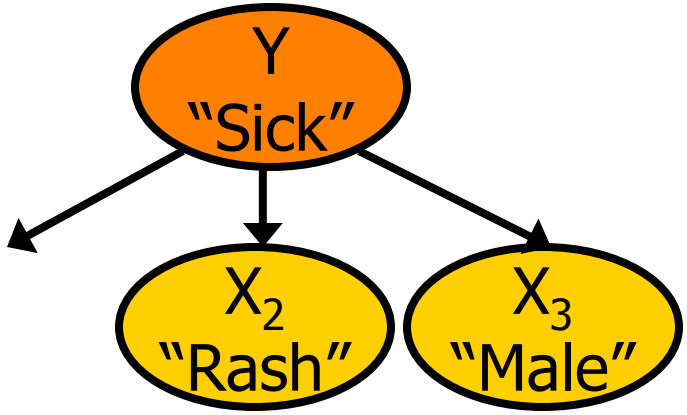
$A := A \cup \{s^*\}$

How well can this simple heuristic do?

Key property: Diminishing returns

Selection A = {}

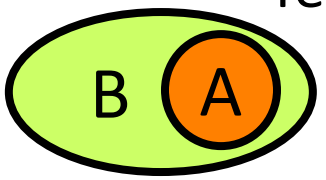
Selection B = $\{X_2, X_3\}$

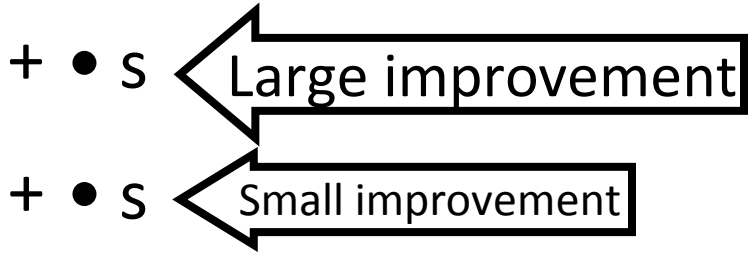


Theorem [Krause, Guestrin UAI '05]**: Information gain F(A) in Naïve Bayes models is submodular!**

feature X_1

Submodularity:





For $A \subseteq B$, $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$

Why is submodularity useful?

Theorem [Nemhauser et al '78]

Greedy maximization algorithm returns A_{greedy}:

$$F(A_{greedy}) \geq \underbrace{(1-1/e)}_{\sim 63\%} \max_{|A| \leq k} F(A)$$

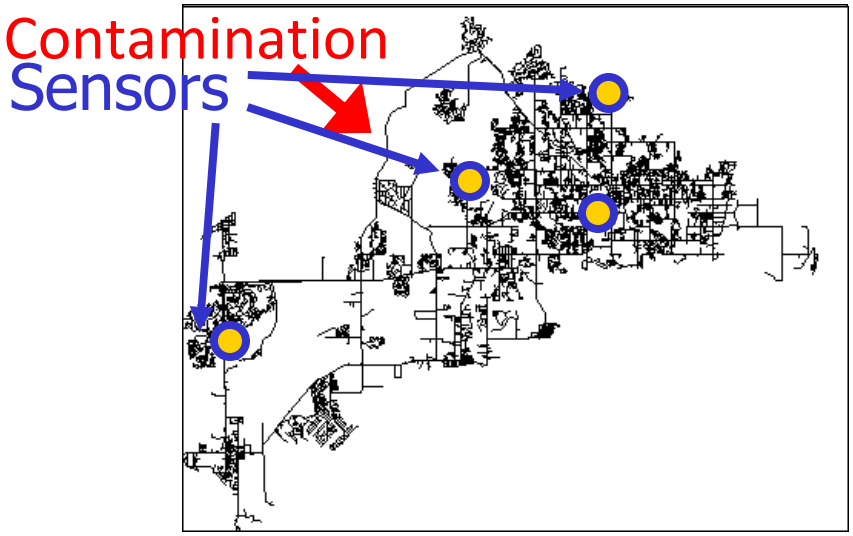
- Greedy algorithm gives near-optimal solution!
- For info-gain: Guarantees best possible unless P = NP! [Krause, Guestrin UAI '05]

Submodularity is an incredibly useful and powerful concept!

Monitoring water networks

[Krause et al, J Wat Res Mgt 2008]

- Contamination of drinking water could affect millions of people



Simulator from EPA

Hach Sensor

~$14K

- Place sensors to detect contaminations
- "Battle of the Water Sensor Networks" competition

Where should we place sensors to quickly detect contamination?

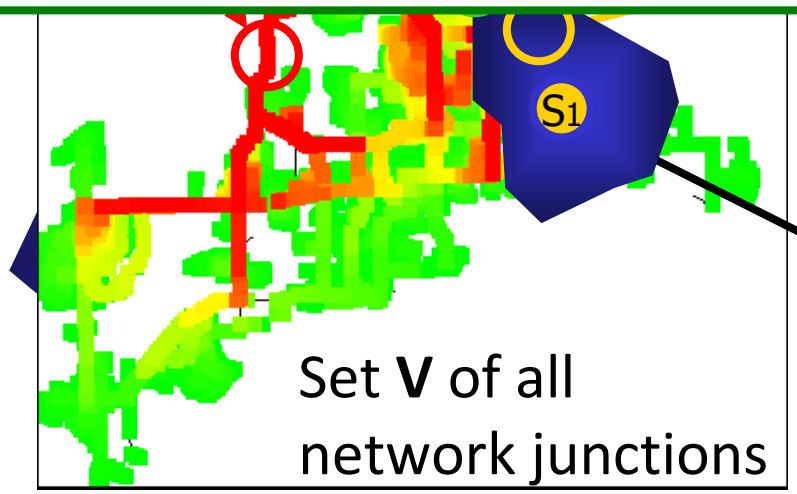
Model-based sensing

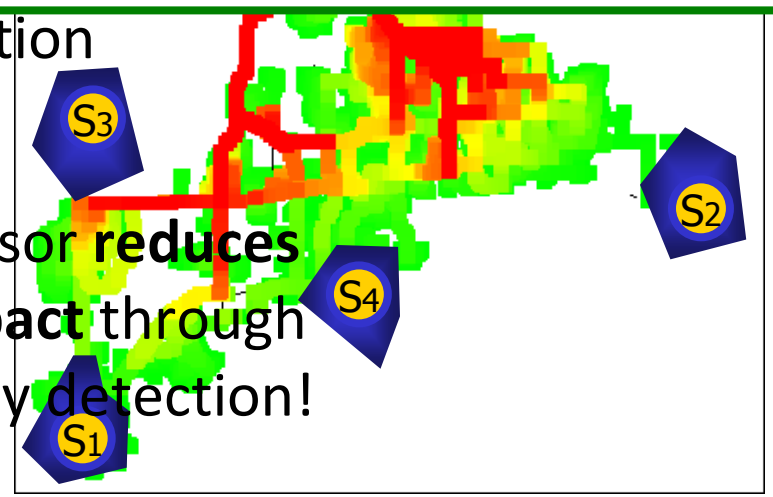
- Utility of placing sensors based on model of the world
  - For water networks: Water flow simulator from EPA
- F(A)=Expected impact reduction placing sensors at A

Model predicts Low impact location

Theorem [Krause et al., J Wat Res Mgt '08]**:**

Impact reduction F(A) in water networks is submodular!





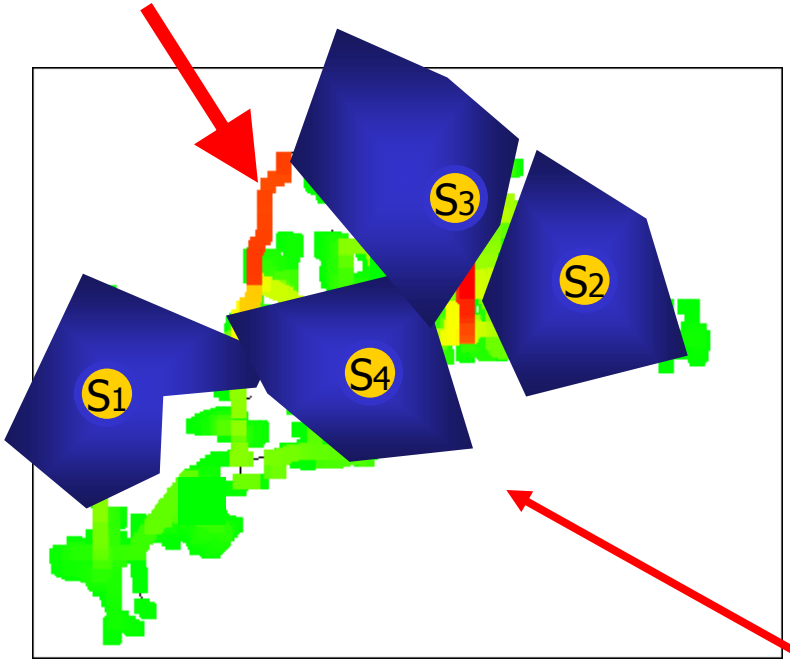
High impact reduction F(A) = 0.9 Low impact reduction F(A)=0.01

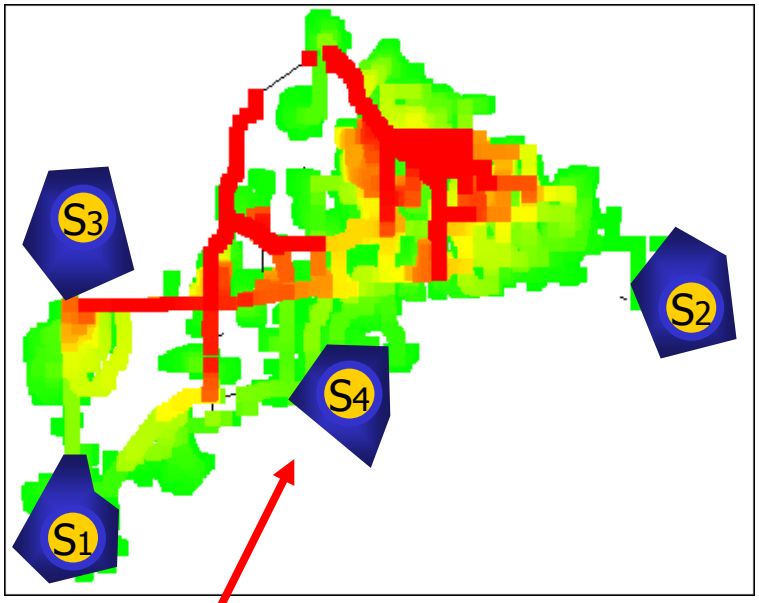
Battle of the Water Sensor Networks Competition

- Real metropolitan area network (12,527 nodes)
- Water flow simulator provided by EPA
- 3.6 million contamination events
- Multiple objectives:
  - Detection time, affected population, …
- Place sensors that detect well "on average"

What about worst-case?
[Krause et al., NIPS '07]

Knowing the sensor locations, an adversary contaminates here!





Placement detects well on **"average-case"** (accidental) contamination

Very different average-case impact, **Same** worst-case impact

Where should we place sensors to quickly detect in the **worst case**?

Constrained maximization: Outline

Utility function

Selected set

$$\max_{\mathcal{A} \subseteq \mathcal{V}} F(\mathcal{A})$$

Selection cost

Budget

$$\text{subject to } C(\mathcal{A}) \leq B$$

Subset selection ✓

Robust optimization

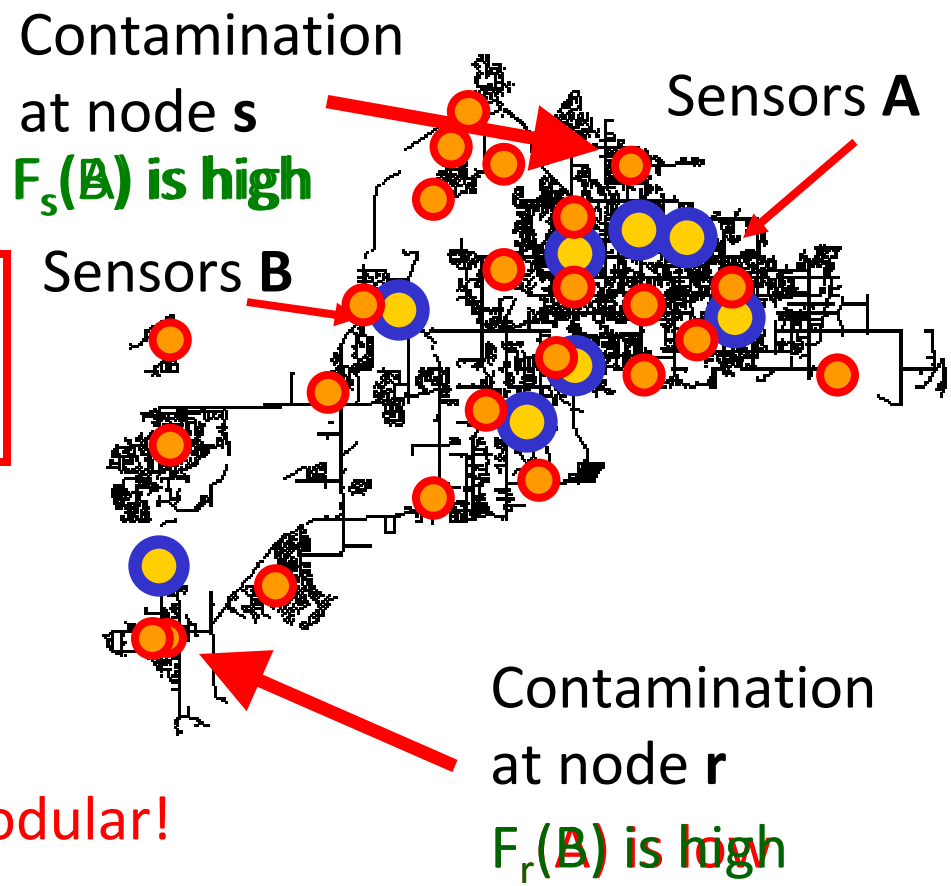
Complex constraints

Optimizing for the worst case

- Separate utility function F_i for each contamination i
- $F_i(A)$ = impact reduction by sensors A for contamination i

Want to solve

$$\boxed{\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A})}$$



Each of the F_i is submodular

Unfortunately, $\min_i F_i$ not submodular!

How can we solve this **robust optimization** problem?

How does the greedy algorithm do?

V={  ,  ,  }

Can only buy k=2

Greedy picks  first

Optimal solution

Then, can choose only  or 

Hence we can't find **any** approximation algorithm.

Or can we?

Optimal score: 1

Greedy score: ε

➔ Greedy does arbitrarily badly. Is there something better?

Theorem: The problem $\max_{|A| \leq k} \min_i F_i(A)$ does not admit **any** approximation unless **P=NP**

Alternative formulation

If somebody told us the optimal value,

$$c^* = \max_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A})$$

can we recover the optimal solution A^*?

How does this help to solve $\max_{|A| \leq k} \min_i F_i(A)$

Need to find

$$\mathcal{A}^* = \operatorname*{argmin}_{\mathcal{A}} |\mathcal{A}| \text{ such that } \min_i F_i(\mathcal{A}) \geq c^*$$

$0 \leq c^* \leq \min_i F_i(V) = c_{max}$

$[\quad | \quad]$

$0 \qquad c_{max}$

Is this any easier?

Yes, if we relax the constraint $|A| \leq k$

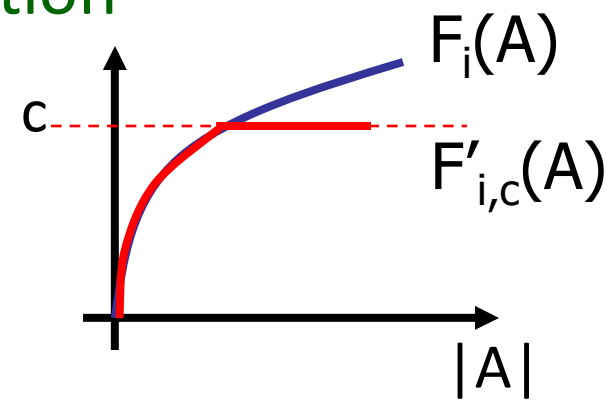
Solving the alternative problem

Trick: For each F_i and c, define truncation

Remains submodular!

$$F'_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), c\}$$

$$F'_{\text{avg},c}(\mathcal{A}) = \frac{1}{m}\sum_i F'_{i,c}(\mathcal{A})$$



Problem 1 (last slide)

$$\min_{\mathcal{A}} |\mathcal{A}|$$
$$\text{s.t.} \quad \boxed{\min_i F_i(\mathcal{A})} \geq c$$

Non-submodular ☹

Don't know how to solve

Problem 2

$$\min_{\mathcal{A}} |\mathcal{A}|$$
$$\text{s.t.} \quad \boxed{F'_{\text{avg},c}(\mathcal{A})} \geq c$$

Submodular!

But appears as constraint?

Same optimal solutions!

Solving one solves the other

Maximization vs. coverage

Previously: Wanted

$$A^* = \text{argmax } F(A) \text{ s.t. } |A| \leq k$$

Now need to solve:

$$A^* = \text{argmin } |A| \text{ s.t. } F(A) \geq Q$$

<u>Greedy algorithm:</u>

Start with $A := \emptyset$;

While $F(A) < Q$ and $|A| < n$

$s^* := \text{argmax}_s F(A \cup \{s\})$

$A := A \cup \{s^*\}$

For bound, assume F is integral.
If not, just round it.

Theorem [Wolsey et al]: Greedy will return A_{greedy}

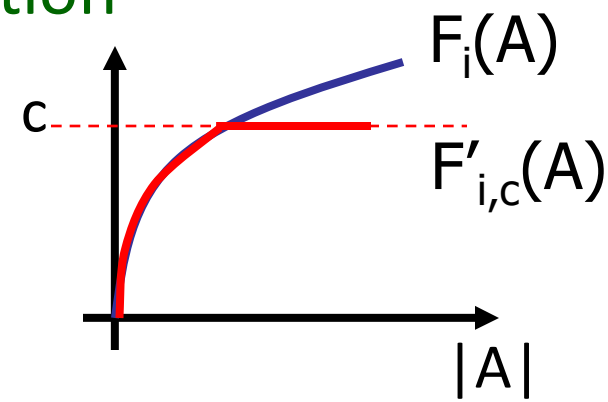
$$|A_{greedy}| \leq (1+\log \max_s F(\{s\}))\ |A_{opt}|$$

Solving the alternative problem

Trick: For each F_i and c, define truncation

$$F'_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), c\}$$

$$F'_{\text{avg},c}(\mathcal{A}) = \frac{1}{m}\sum_i F'_{i,c}(\mathcal{A})$$



Problem 1 (last slide)

$$\min_{\mathcal{A}} |\mathcal{A}|$$
$$\text{s.t.} \quad \boxed{\min_i F_i(\mathcal{A})} \geq c$$

Non-submodular ☹

Don't know how to solve

Problem 2

$$\min_{\mathcal{A}} |\mathcal{A}|$$
$$\text{s.t.} \quad \boxed{F'_{\text{avg},c}(\mathcal{A})} \geq c$$

Submodular!

Can use greedy algorithm!

Back to our example

- Guess c=1
- First pick 
- Then pick 

➔ Optimal solution!

Set A	F_1	F_2	$\min_i F_i$
	1	0	0
	0	2	0
	ε	ε	ε

How do we find c?

Do binary search!

SATURATE Algorithm

[Krause et al, NIPS '07]

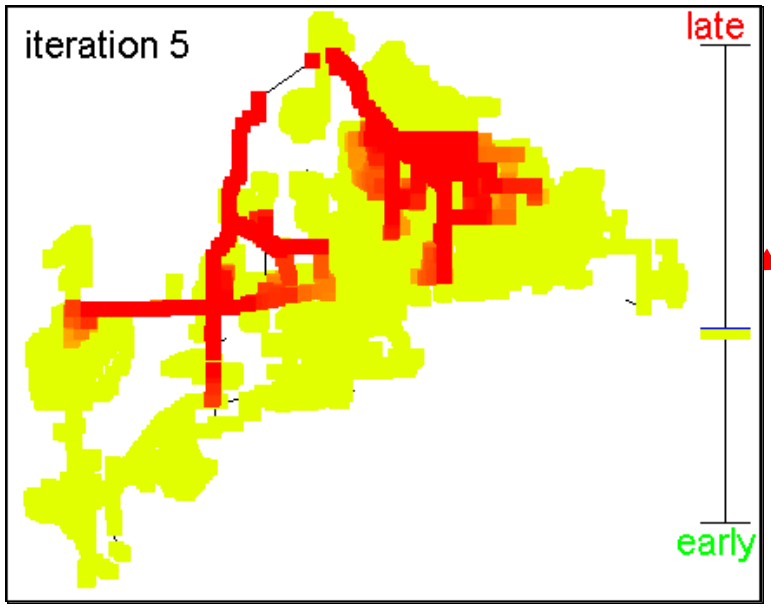
Given: set V, integer k and monotonic SFs $F_1,\ldots,F_m$

Initialize $c_{min}=0$, $c_{max} = \min_i F_i(V)$

Do binary search: $c = (c_{min}+c_{max})/2$

- Greedily find A_G such that $F'_{avg,c}(A_G) = c$
- If $|A_G| \leq \alpha k$: increase c_{min}
- If $|A_G| > \alpha k$: decrease c_{max}

until convergence



Truncation threshold (color)

Theoretical guarantees

[Krause et al, NIPS '07]

Theorem: The problem $\max_{|A| \leq k} \min_i F_i(A)$ does not admit **any** approximation unless **P=NP** ☹

Theorem: *SATURATE* finds a solution A_S such that

$$\min_i F_i(A_S) \geq OPT_k \text{ and } |A_S| \leq \alpha k$$

where $OPT_k = \max_{|A| \leq k} \min_i F_i(A)$

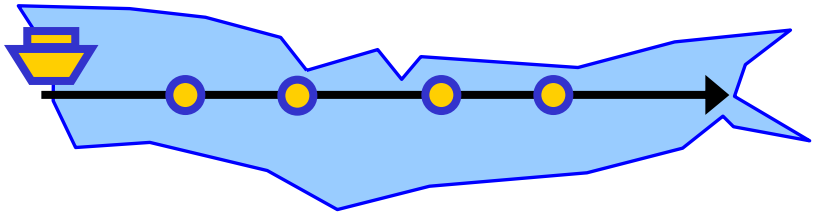
$\alpha = 1 + \log \max_s \sum_i F_i(\{s\})$

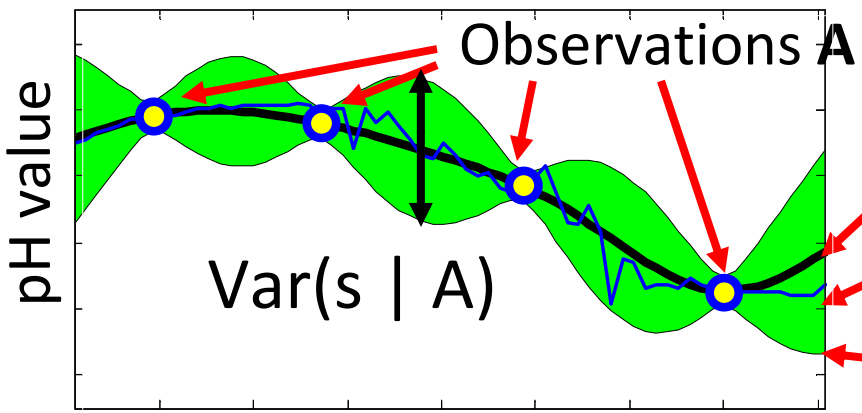
Theorem:

If there were a polytime algorithm with better factor $\beta < \alpha$, then $NP \subseteq DTIME(n^{\log \log n})$

Example: Lake monitor

- Monitor pH values using robotic sensor transect





Prediction at unobserved locations

True (hidden) pH values

Use **probabilistic model** (Gaussian processes) to estimate prediction error

Where should we sense to **minimize our maximum error**?

$$\min_s \underbrace{\mathrm{Var}(s) - \mathrm{Var}(s \mid \mathcal{A})}_{\text{(often) submodular}}$$

[Das & Kempe '08]

➔ Robust submodular optimization problem!

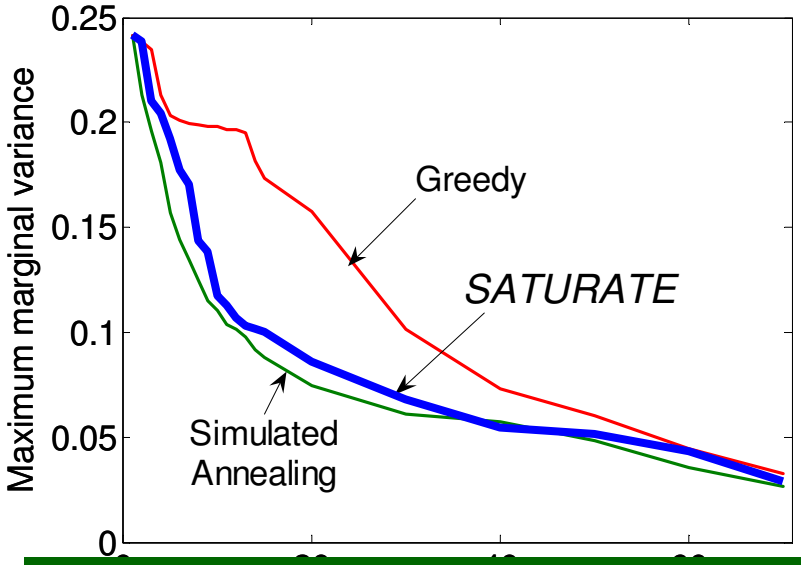
Comparison with state of the art

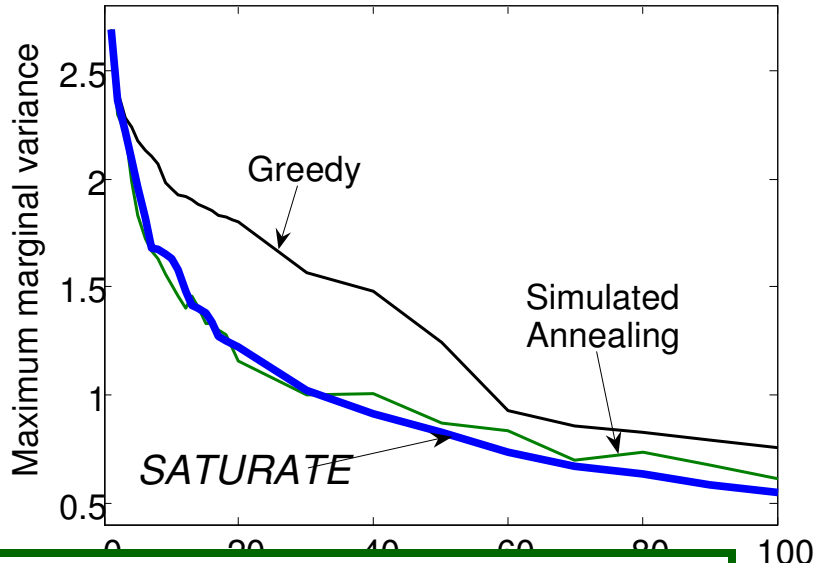
Algorithm used in geostatistics: *Simulated Annealing*

[Sacks & Schiller '88, van Groeningen & Stein '98, Wiens '05,…]

7 parameters that need to be fine-tuned

better

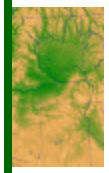




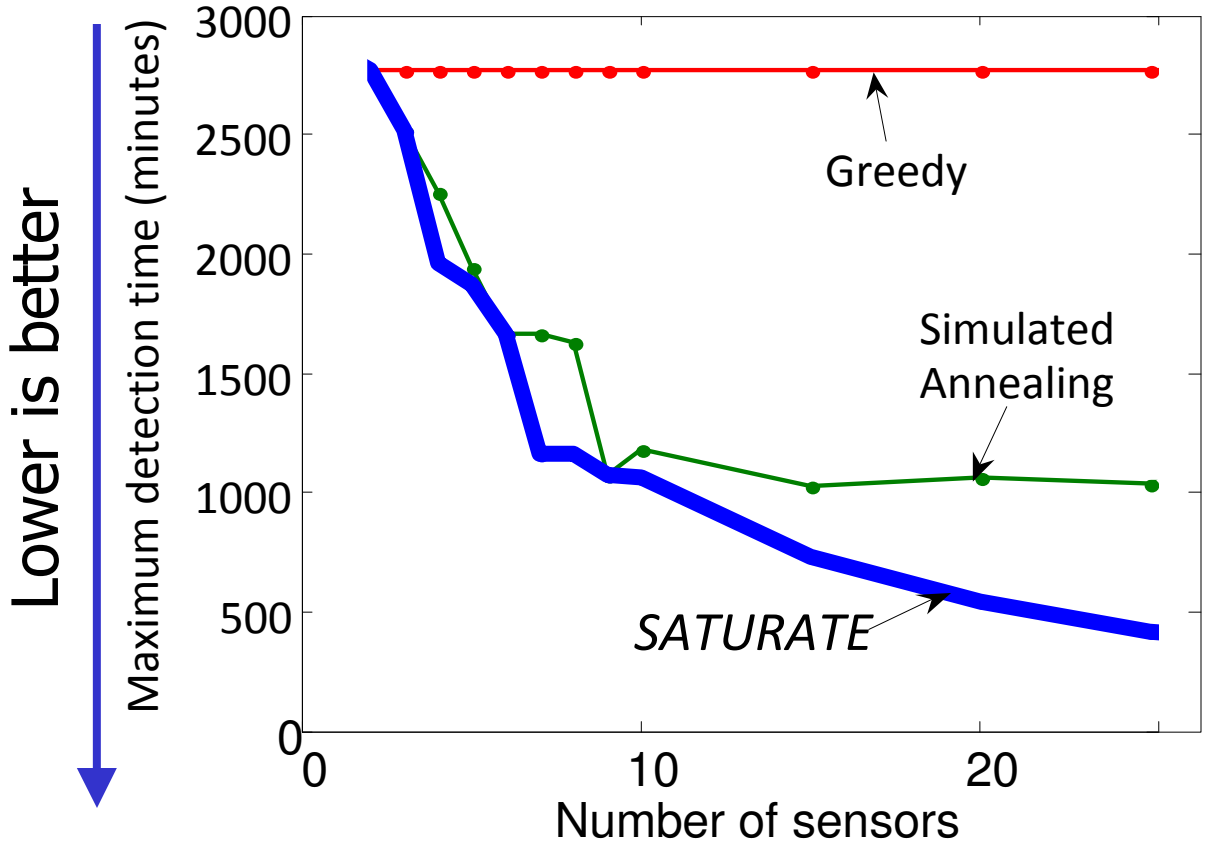
Envi

SATURATE is competitive & 10x faster

No parameters to tune!



Results on water networks



No decrease until **all** contaminations detected!

Water networks

60% lower worst-case detection time!

Worst- vs. average case

Given: Set V, submodular functions $F_1,\ldots,F_m$

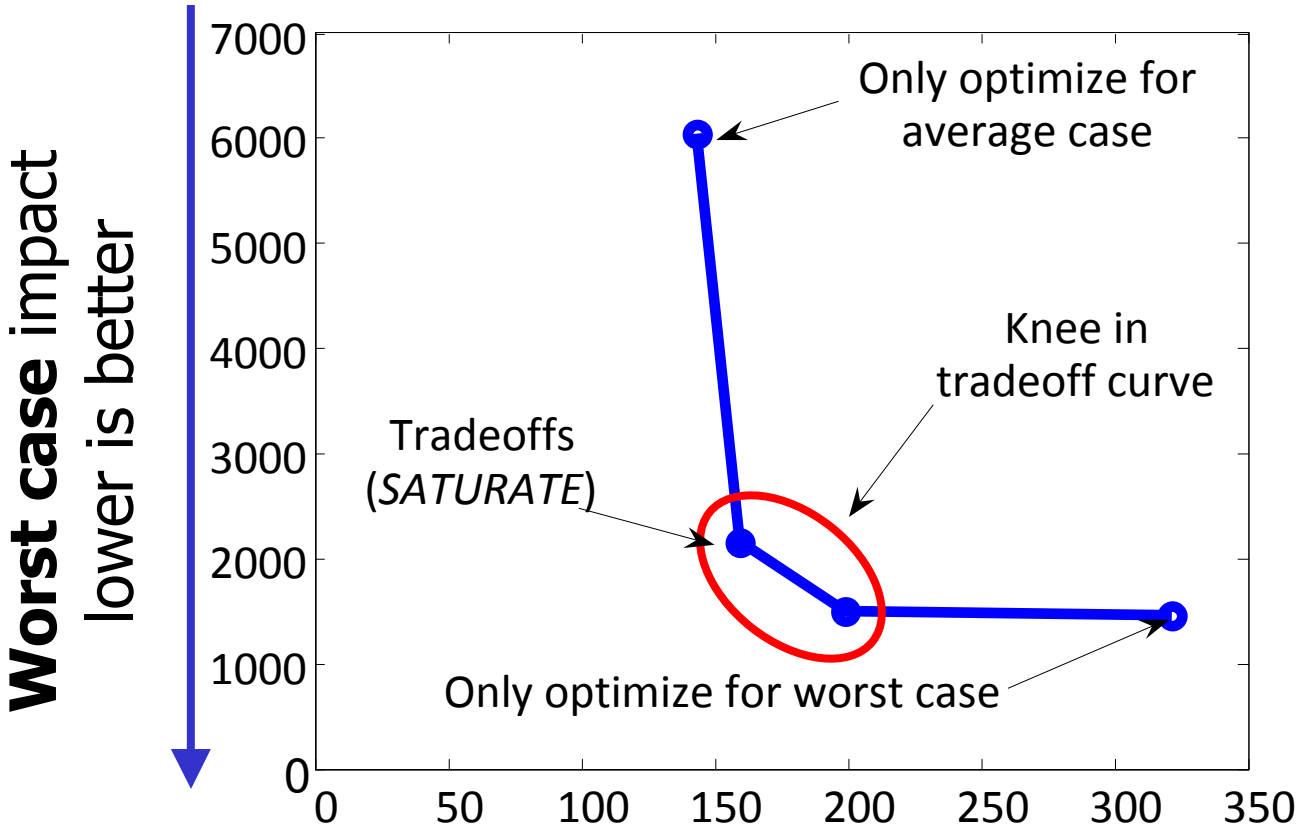
Average-case score	Worst-case score
$F_{ac}(\mathcal{A}) = \frac{1}{m}\sum_i F_i(\mathcal{A})$	$F_{wc}(\mathcal{A}) = \min_i F_i(\mathcal{A})$

Want to optimize **both** average- and worst-case score!

Can modify *SATURATE* to solve this problem! ☺

- Want: $F_{ac}(A) \geq c_{ac}$ and $F_{wc}(A) \geq c_{wc}$
- Truncate: $\min\{F_{ac}(A),c_{ac}\} + \min\{F_{wc}(A),c_{wc}\} \geq c_{ac}+c_{wc}$

Worst- vs. average case



Water networks data

Can find good compromise between average- and worst-case score!

Constrained maximization: Outline

Utility function

Selected set

$$\max_{\mathcal{A} \subseteq \mathcal{V}} F(\mathcal{A})$$

Selection cost

Budget

$$\text{subject to } C(\mathcal{A}) \leq B$$

Subset selection ✓

Robust optimization ✓

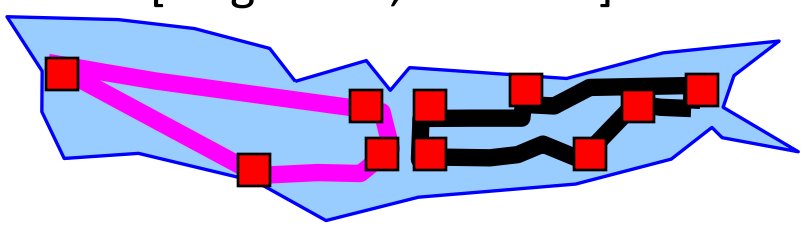
Complex constraints

Other aspects: Complex constraints

$\max_{\mathbf{A}} F(\mathbf{A})$ or $\max_{\mathbf{A}} \min_i F_i(\mathbf{A})$ subject to

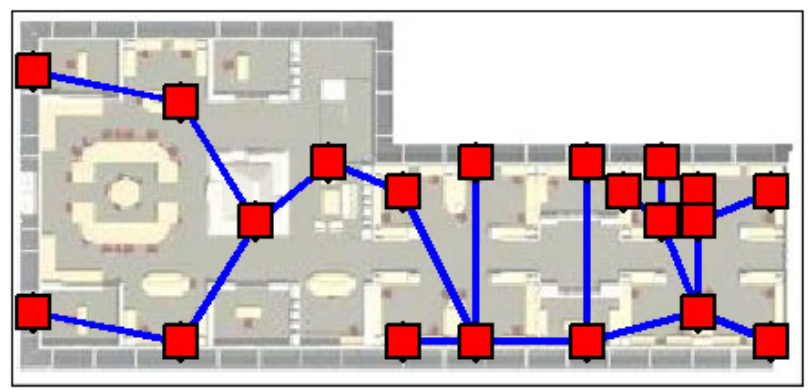
- So far: $|\mathbf{A}| \leq k$
- In practice, more complex constraints:
- Different costs: $C(A) \leq B$

Locations need to be connected by paths
[Chekuri & Pal, FOCS '05]
[Singh et al, IJCAI '07]



Lake monitoring

Sensors need to communicate (form a routing tree)



Building monitoring

Non-constant cost functions

- For each $s \in V$, let $c(s)>0$ be its cost (e.g., feature acquisition costs, …)
- Cost of a set $C(A) = \sum_{s \in A} c(s)$ (modular function!)
- Want to solve

$$A^* = \text{argmax } F(A) \text{ s.t. } C(A) \leq B$$

<u>Cost-benefit greedy algorithm:</u>

Start with $A := \emptyset$;

While there is an $s \in V \setminus A$ s.t. $C(A \cup \{s\}) \cdot B$

$$s^* = \underset{s : C(\mathcal{A} \cup \{s\}) \leq B}{\operatorname{argmax}} \frac{F(\mathcal{A} \cup \{s\}) - F(\mathcal{A})}{c(s)}$$

$A := A \cup \{s^*\}$

Performance of cost-benefit greedy

Want

$\max_A F(A)$ s.t. $C(A) \leq 1$

Set A	F(A)	C(A)
{a}	2ε	ε
{b}	1	1

Cost-benefit greedy picks a.

Then cannot afford b!

➔ Cost-benefit greedy performs arbitrarily badly!

Cost-benefit optimization
[Wolsey '82, Sviridenko '04, Leskovec et al '07]

Theorem

- A_{CB}: cost-benefit greedy solution and
- A_{UC}: unit-cost greedy solution (i.e., ignore costs)

Then

$$\max \{ F(A_{CB}), F(A_{UC}) \} \geq ½ (1-1/e) \text{ OPT}$$

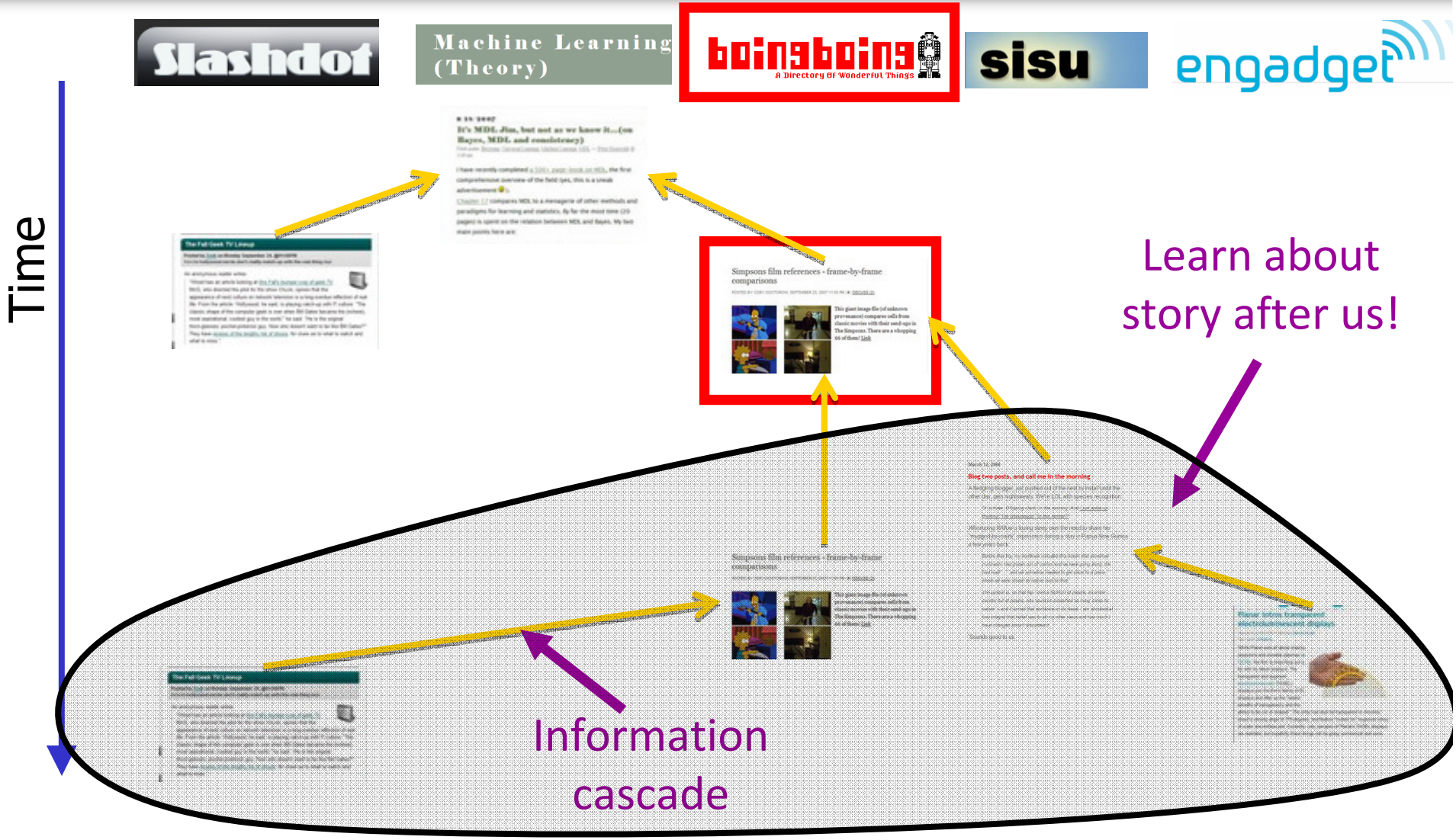
Can still compute online bounds and speed up using lazy evaluations

Note: Can also get

- (1-1/e) approximation in time $O(n^4)$ [Sviridenko '04]
- Slightly better than ½ (1-1/e) in $O(n^2)$ [Wolsey '82]

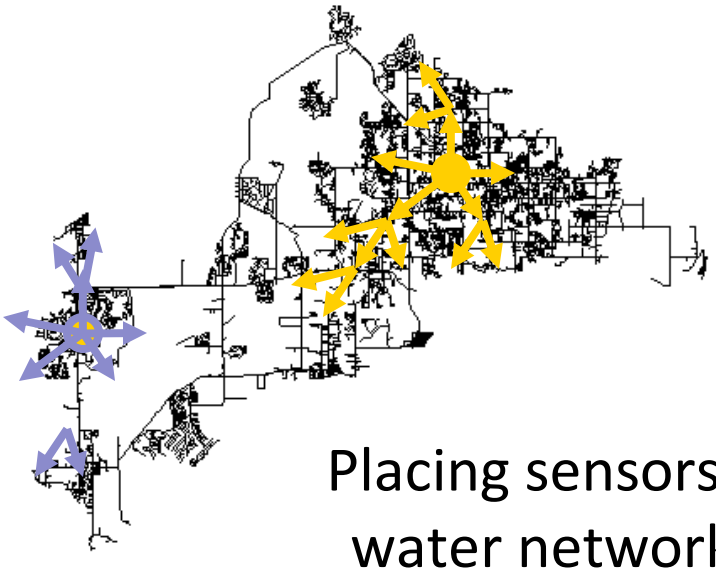
Example: Cascades in the Blogosphere

[Leskovec, Krause, Guestrin, Faloutsos, VanBriesen, Glance '07]



Which blogs should we read to learn about big cascades early?

Water vs. Web



Placing sensors in water networks

vs.

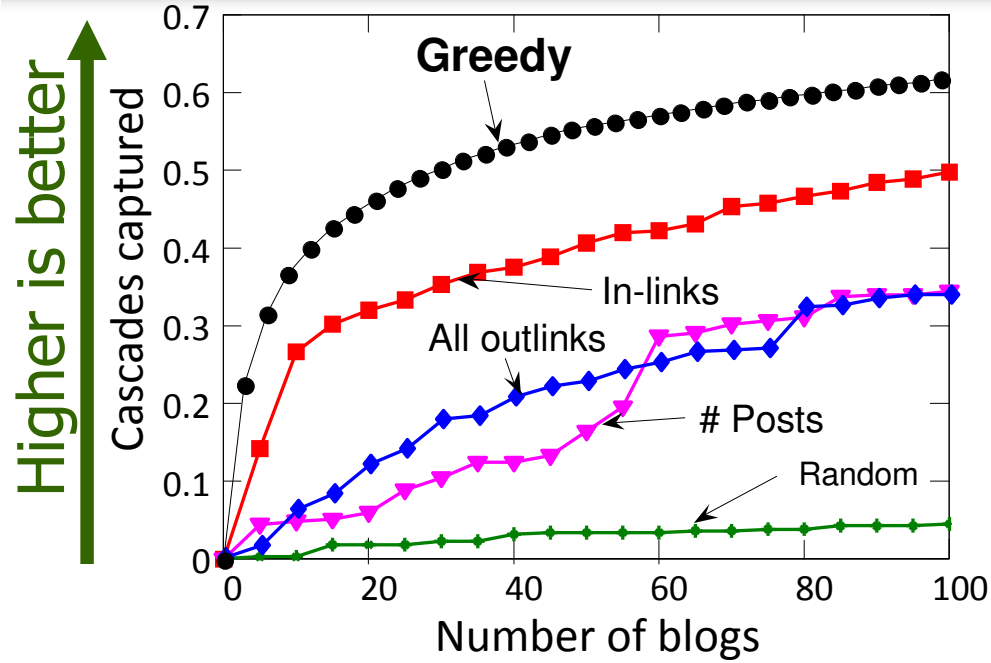
Selecting informative blogs

- In both problems we are given
  - Graph with nodes (junctions / blogs) and edges (pipes / links)
  - Cascades spreading dynamically over the graph (contamination / citations)
- Want to pick nodes to **detect big cascades early**

In both applications, utility functions submodular ☺

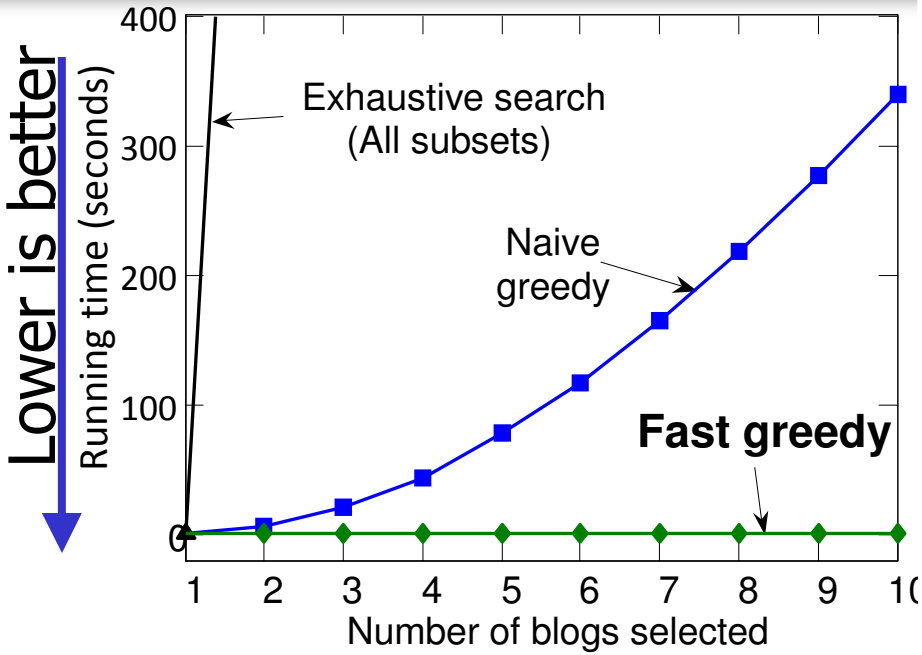
[Generalizes Kempe et al, KDD '03]

Performance on Blog selection



Blog selection

~45k blogs



Blog selection

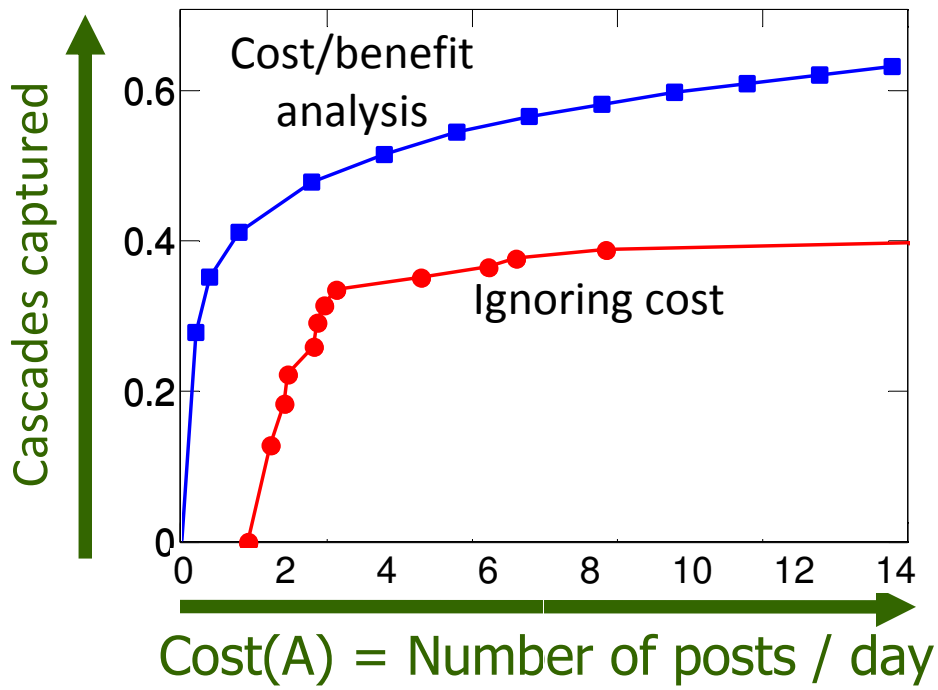
Outperforms state-of-the-art heuristics

700x speedup using submodularity!

Cost of reading a blog

skip

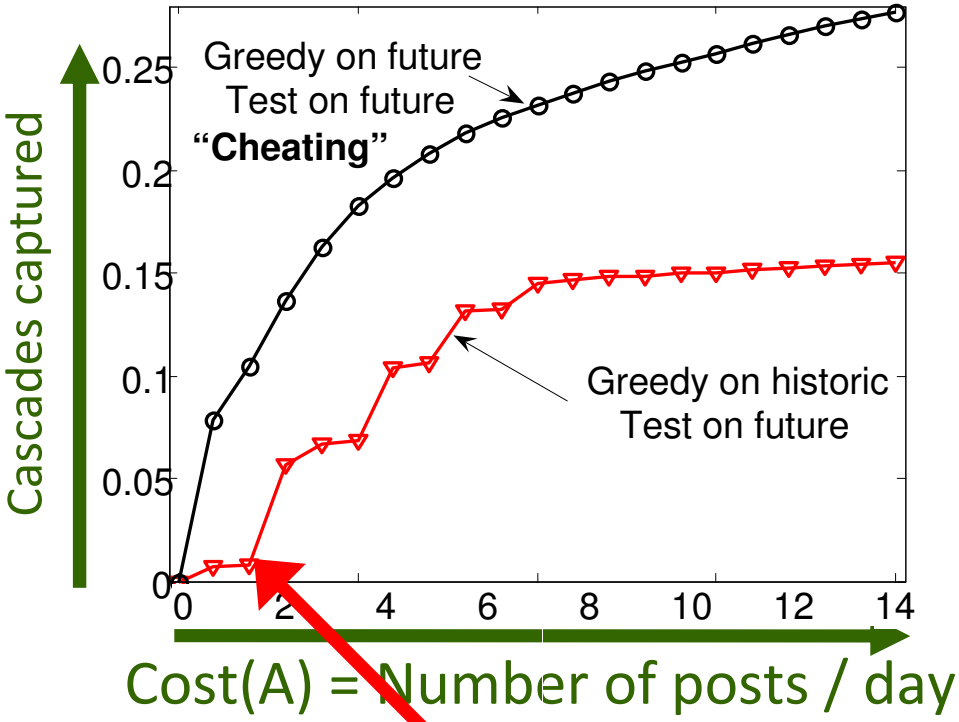
- Naïve approach: Just pick 10 best blogs
- Selects big, well known blogs (Instapundit, etc.)
- These contain many posts, take long to read!



Cost-benefit optimization picks summarizer blogs!

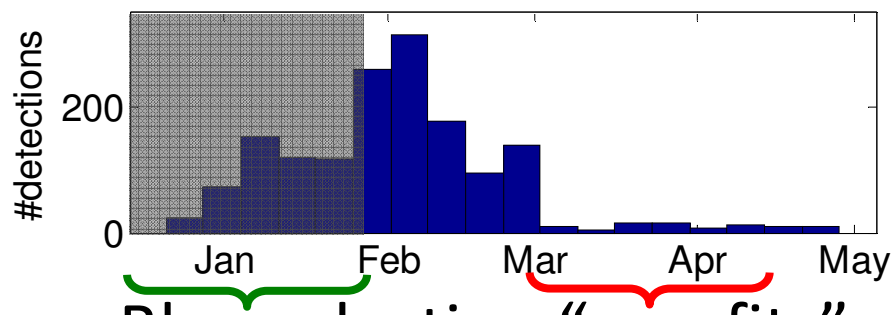
Predicting the "hot" blogs

- Want blogs that will be informative in the future
- Split data set; train on historic, test on future



Let's see what goes wrong here.

Detects on training set



Blog selection "overfits" to training data!

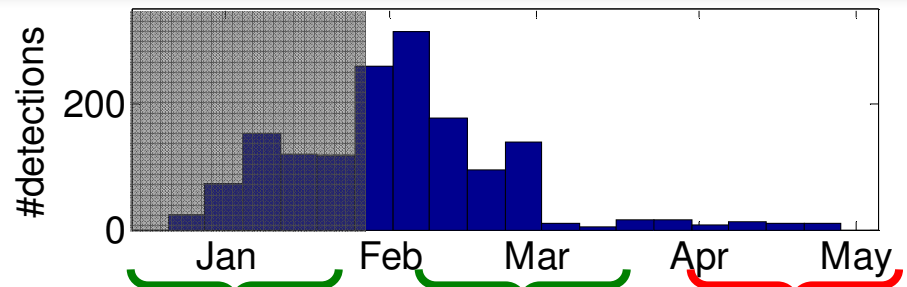
Poor generalization!

Why?

Want blogs that continue to do well!

Robust optimization

"Overfit" blog selection **A**



$F_i(A)$ = detections in interval i

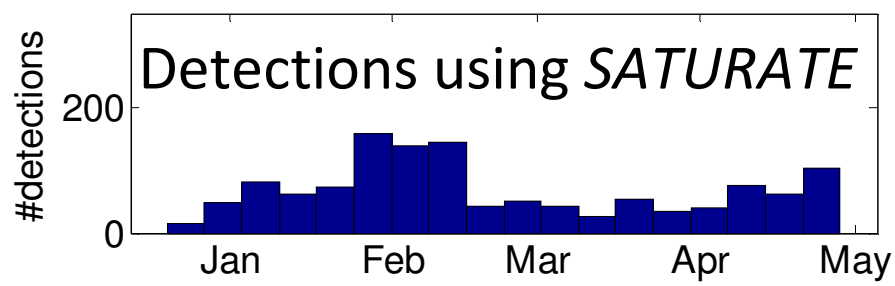
$F_1(A)=.5$ $F_3(A)=.6$ $F_5(A)=.02$

$F_2(A)=.8$ $F_4(A)=.01$

Optimize worst-case

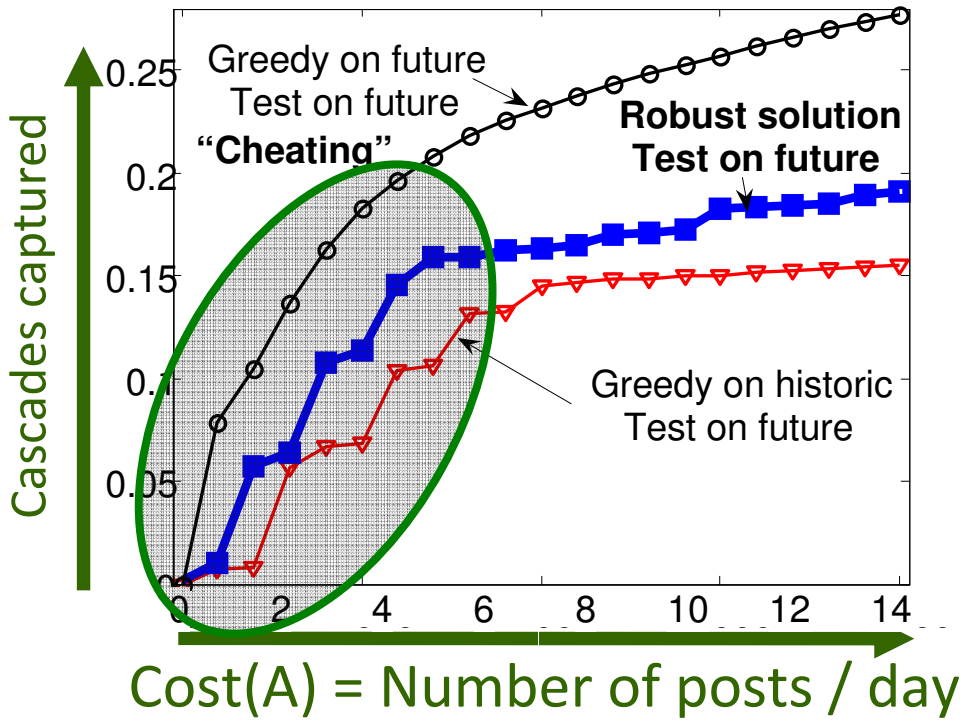
$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A})$$

"Robust" blog selection **A***



Robust optimization ⇔ Regularization!

Predicting the "hot" blogs



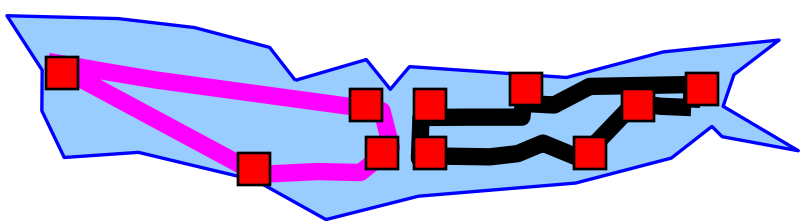
50% better generalization!

Other aspects: Complex constraints skip

$\max_{\mathbf{A}} F(\mathbf{A})$ or $\max_{\mathbf{A}} \min_i F_i(\mathbf{A})$ subject to

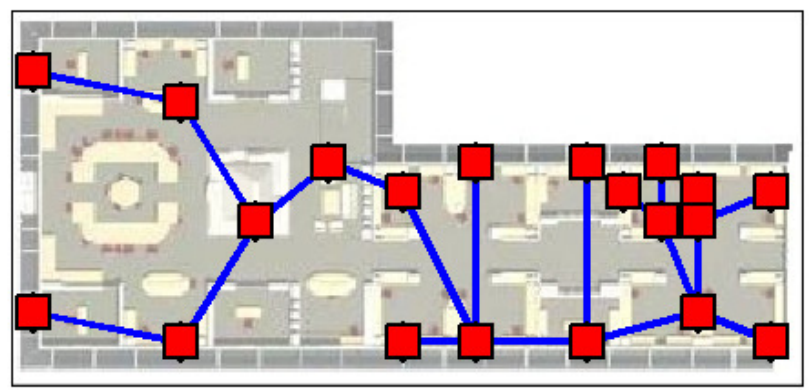
- So far: $|\mathbf{A}| \leq k$
- In practice, more complex constraints:
- Different costs: $C(A) \leq B$

Locations need to be connected by paths
[Chekuri & Pal, FOCS '05]
[Singh et al, IJCAI '07]



Lake monitoring

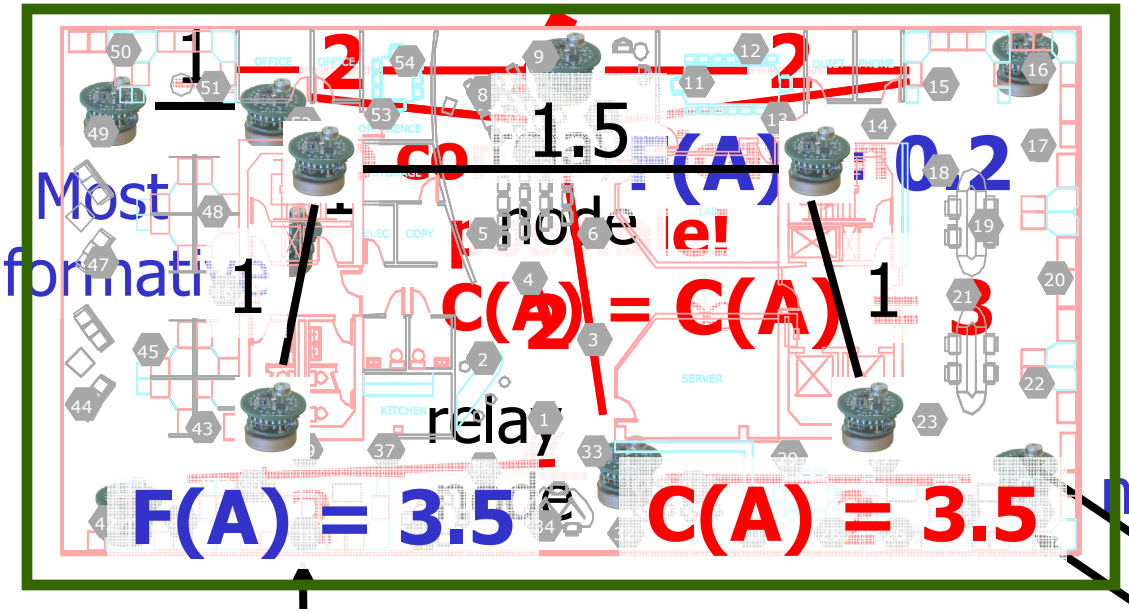
Sensors need to communicate (form a routing tree)



Building monitoring

Naïve approach: Greedy-connect

- Simple heuristic: **Greedily** optimize submodular utility function F(A)
- *Then* **add** nodes to minimize communication cost C(A)



Most informati

F(A) = 4

C(A) = 10

Second most informative

efficient

Very informative, High communication cost! ☹

Communication cost = Expected # of trials (learned using Gaussian Processes)

Want to find optimal tradeoff between information and communication cost

The **pSPIEL** Algorithm
[Krause, Guestrin, Gupta, Kleinberg IPSN 2006]

- ***pSPIEL***: Efficient **nonmyopic** algorithm

  (**p**added **S**ensor **P**lacements at **I**nformative and cost-**E**ffective **L**ocations)

  - **Decompose** sensing region into small, well-separated clusters
  - Solve cardinality constrained problem **per cluster** (greedy)
  - **Combine** solutions using k-MST algorithm

Guarantees for *pSPIEL*

[Krause, Guestrin, Gupta, Kleinberg IPSN 2006]

Theorem:

pSPIEL finds a tree T with

submodular utility	$F(T) \geq$	$\Omega(1)$	OPT_F
communication cost	$C(T) \leq$	$O(\log \lvert V \rvert)$	OPT_C

What you should know

- Many important objective functions in Bayesian experimental design are monotonic & submodular
  - Entropy
  - Information gain*
  - Variance reduction*
  - Detection likelihood / time
- Greedy algorithm gives near-optimal solution
- Can also solve more complex problems
  - Connectedness-constraints (trees/paths)
  - Robustness

*under certain assumptions