Optimized Information Gathering in Robotics and Sensor Networks

Andreas Krause

California Institute of Technology

rsrg@caltech

..where theory and practice collide

Information gathering problems

- Want to learn something about the state of the world
  - Estimate water quality in a geographic region, detect outbreaks, …
- We can choose (partial) observations

  - Make measurements, place sensors, choose experimental parameters …
- … but they are expensive / limited
  - hardware cost, power consumption, grad student time …

Want to cost-effectively get most useful information!

Related work

Sensing problems considered in

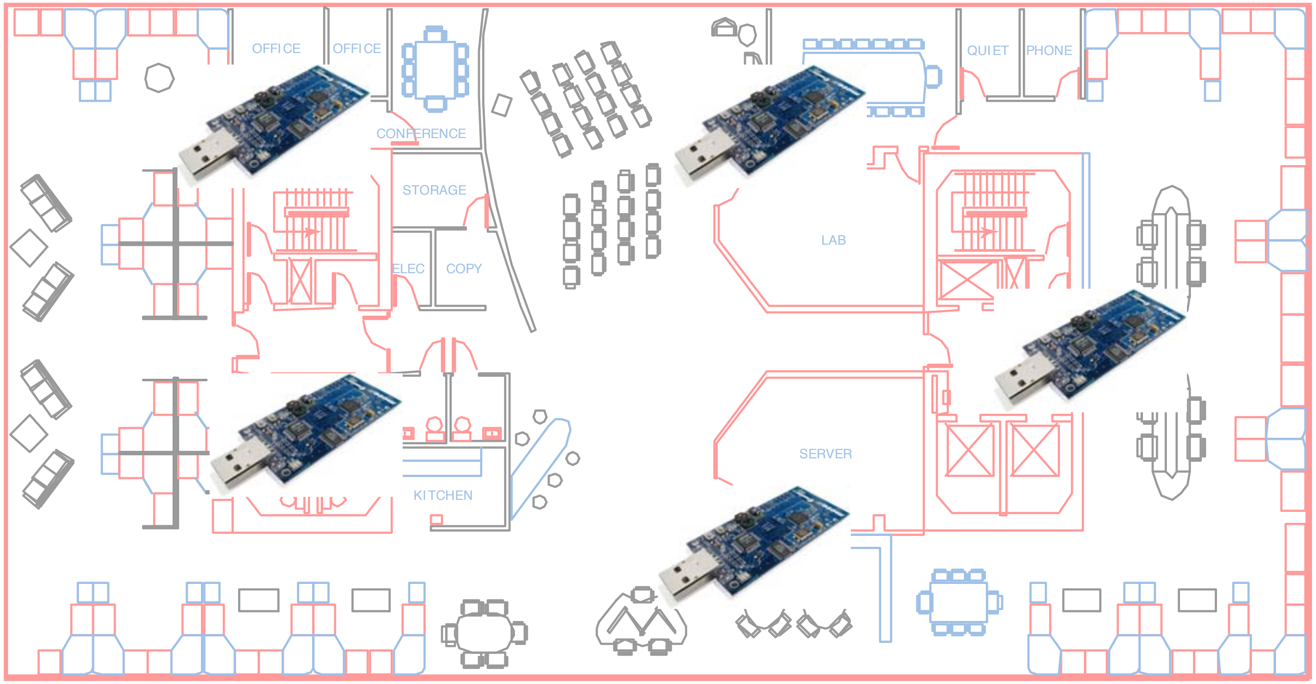
Experimental design (Lindley '56, Robbins '52…), Spatial statistics (Cressie '91, …), Machine Learning (MacKay '92, …), Robotics (Sim&Roy '05, …), Sensor Networks (Zhao et al '04, …), Operations Research (Nemhauser '78, …)

Existing algorithms typically

- **Heuristics**: No guarantees! Can do arbitrarily badly.
- **Find optimal solutions** (Mixed integer programming, POMDPs):
  Very difficult to scale to bigger problems.

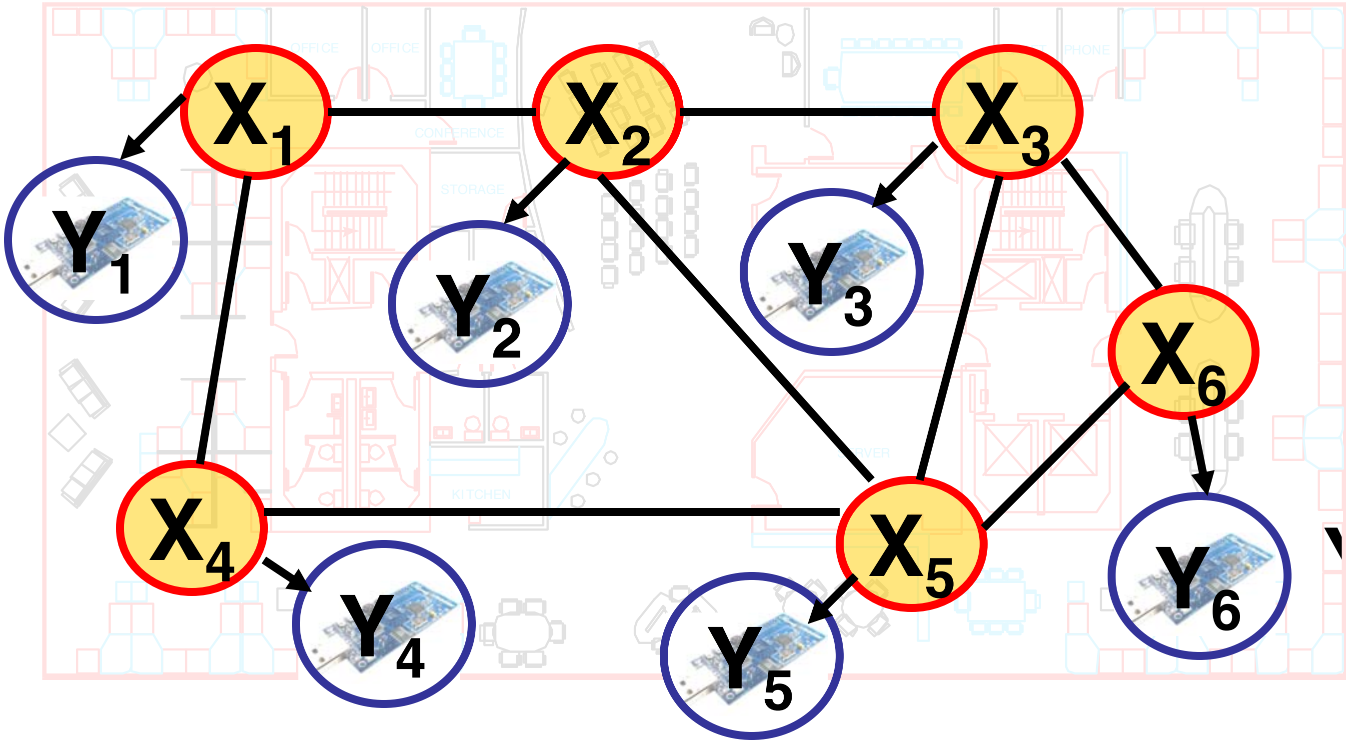
Want algorithms that have theoretical guarantees and scale to large problems!

Running example: Detecting fires



Want to place sensors to detect fires in buildings

Monitoring as Bayesian regression



X_s: temperature at location s

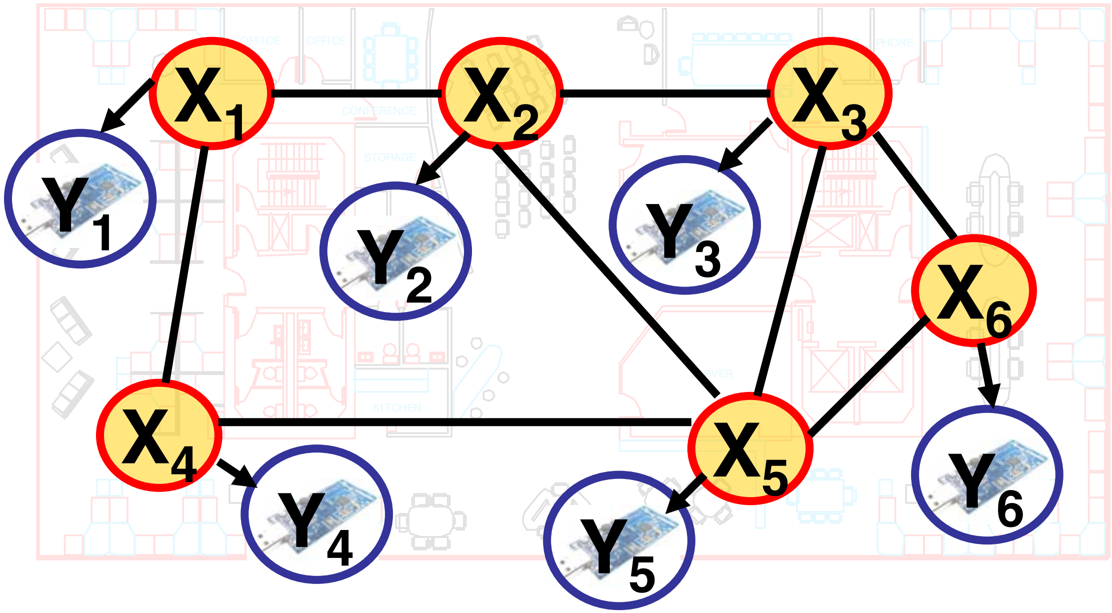
Y_s: sensor value at location s

$Y_s = X_s$ + noise

Joint probability distribution

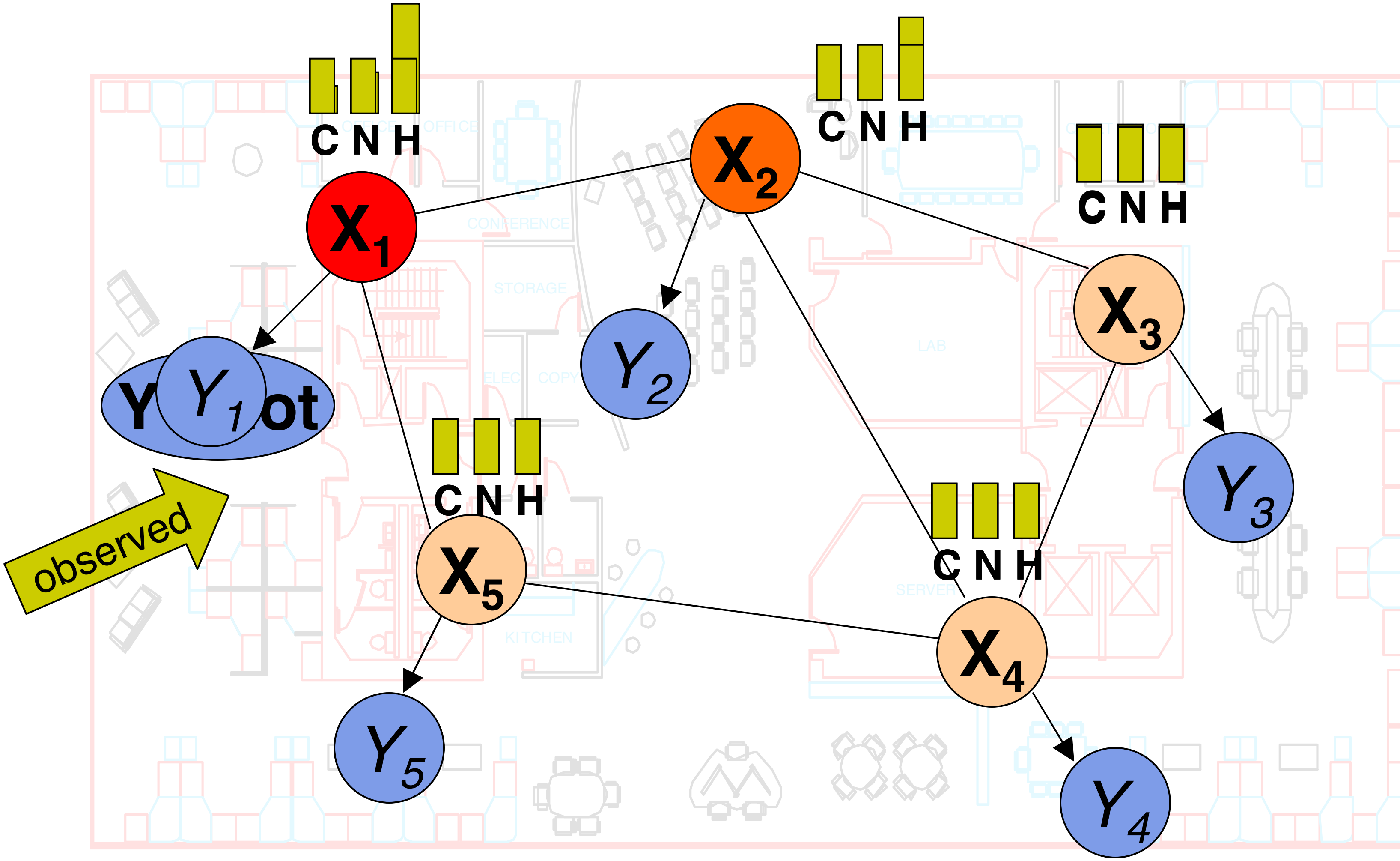
$$\underbrace{P(X_1,\ldots,X_n,Y_1,\ldots,Y_n) = P(X_1,\ldots,X_n)}_{\text{Prior}}\ \underbrace{P(Y_1,\ldots,Y_n \mid X_1,\ldots,X_n)}_{\text{Likelihood}}$$

Why is this useful?



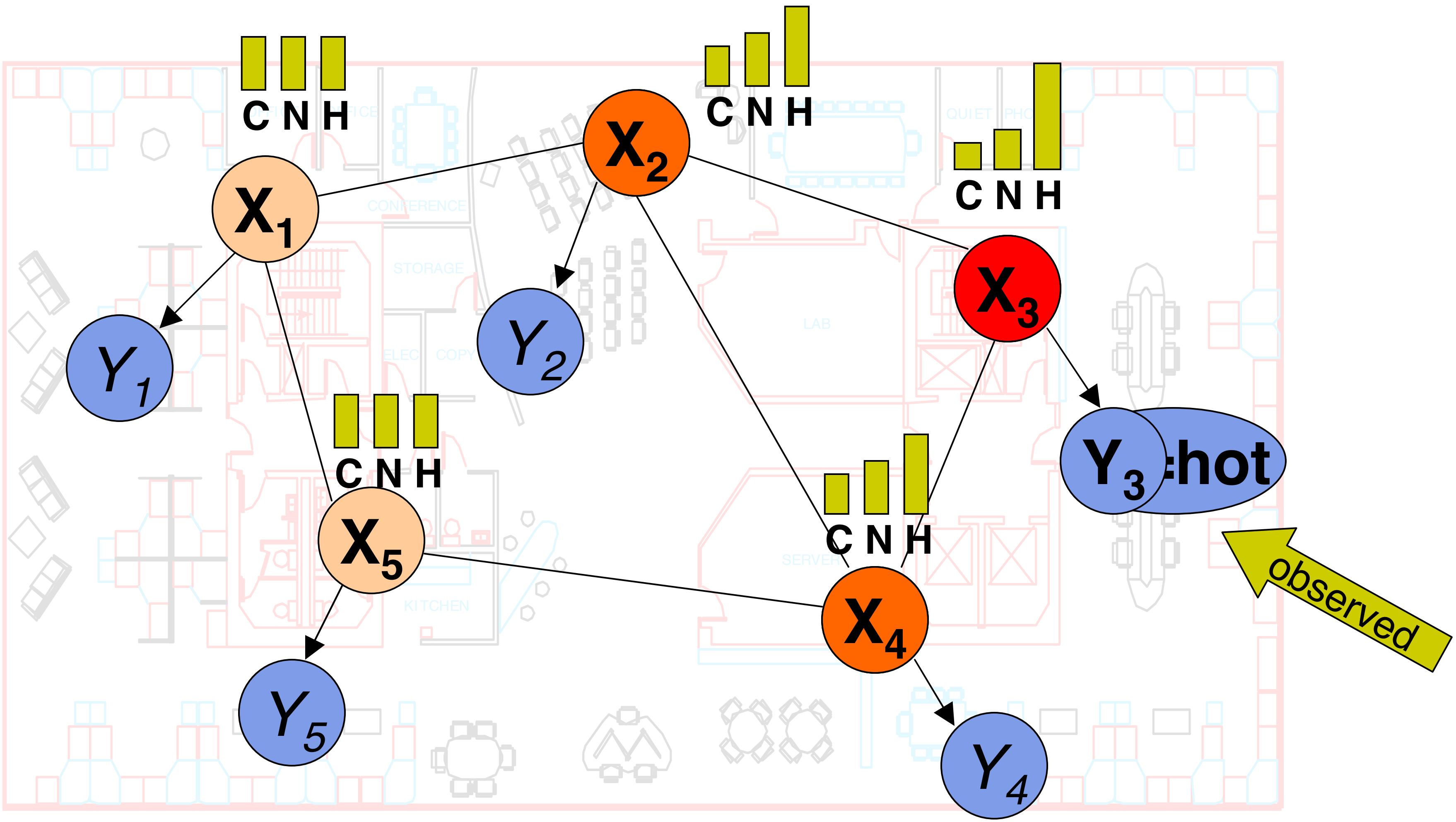
- **Robust reasoning**: Integrate measurements from multiple sensors. E.g.: $P(X_2 \mid y_1, y_2, y_3)$ likely more accurate than $P(X_2 \mid y_2)$
- **Exploiting correlation**: Can predict $P(X_1, X_3 \mid y_2)$
  ➔ Can turn some sensors off to save battery life

Making observations



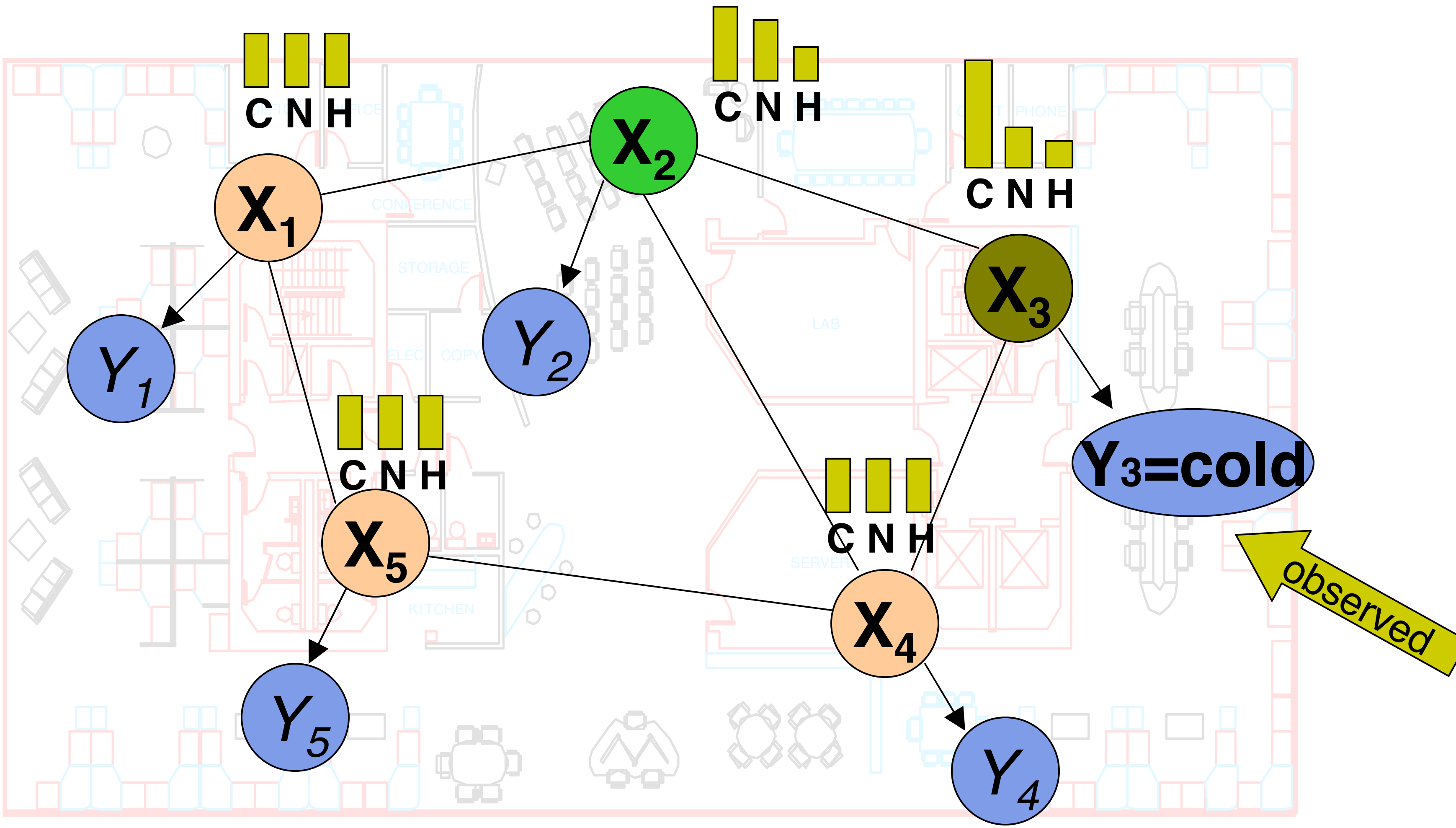
Less uncertain → Reward[P(**X**|Y_1=hot)] = 0.2

Making observations



$$\text{Reward}[\ P(\mathbf{X}|Y_3=\text{hot})] = 0.4$$

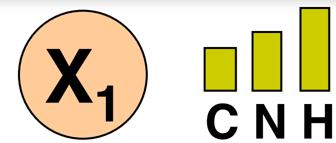
A different outcome…



Reward[P(**X**|Y_3=cold)] = 0.1

Value of information

Should we raise a fire alert?



Temp. X **Actions**	*Fiery hot*	*normal/cold*
No alarm	-$$$	0
Raise alarm	$	-$

Only have belief about temperature P(X = hot | obs)

➔ choose $a^* = \text{argmax}_a \sum_x P(\mathbf{x|obs})\ U(\mathbf{x},a)$

<u>Decision theoretic value of information</u>

$\text{Reward}[\ P(X \mid \text{obs})\ ] = \max_{\mathbf{a}} \sum_x P(\mathbf{x|obs})\ U(\mathbf{x},a)$

Other example reward functions

Entropy

Reward[P(**X**)] = -H(**X**) = $\sum_x P(\mathbf{x}) \log_2 P(\mathbf{x})$

Expected mean squared prediction error (EMSE)

Reward[P(**X**)] = $-1/n \sum_s \mathrm{Var}(X_s)$,

Many other objectives possible and useful…

Value of information [Lindley '56, Howard '64]

For any set A of sensors, its value of information is

$$F(A) = \sum_{\mathbf{y_A}} \underbrace{P(\mathbf{y_A})}_{\substack{\text{Observations} \\ \text{made by sensors } \mathbf{A}}} \underbrace{\mathrm{Reward}[P(\mathbf{X} \mid \mathbf{y_A})]}_{\substack{\text{Reward when observing} \\ Y_A = y_A}}$$

Want to find a set $A^* \mu V$, $|A^*| \cdot k$ s.t.

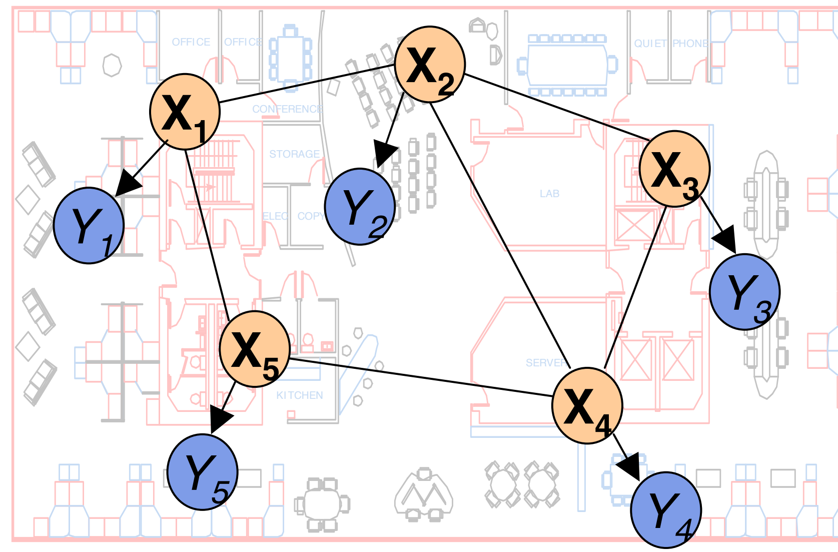
$$A^* = \mathrm{argmax}_{|A| \cdot k} F(A)$$

Optimizing Value of Information

- Given: finite set V of locations
- Want: A*µ V such that

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

Typically NP-hard!



<u>Greedy algorithm:</u>

Start with A = ;

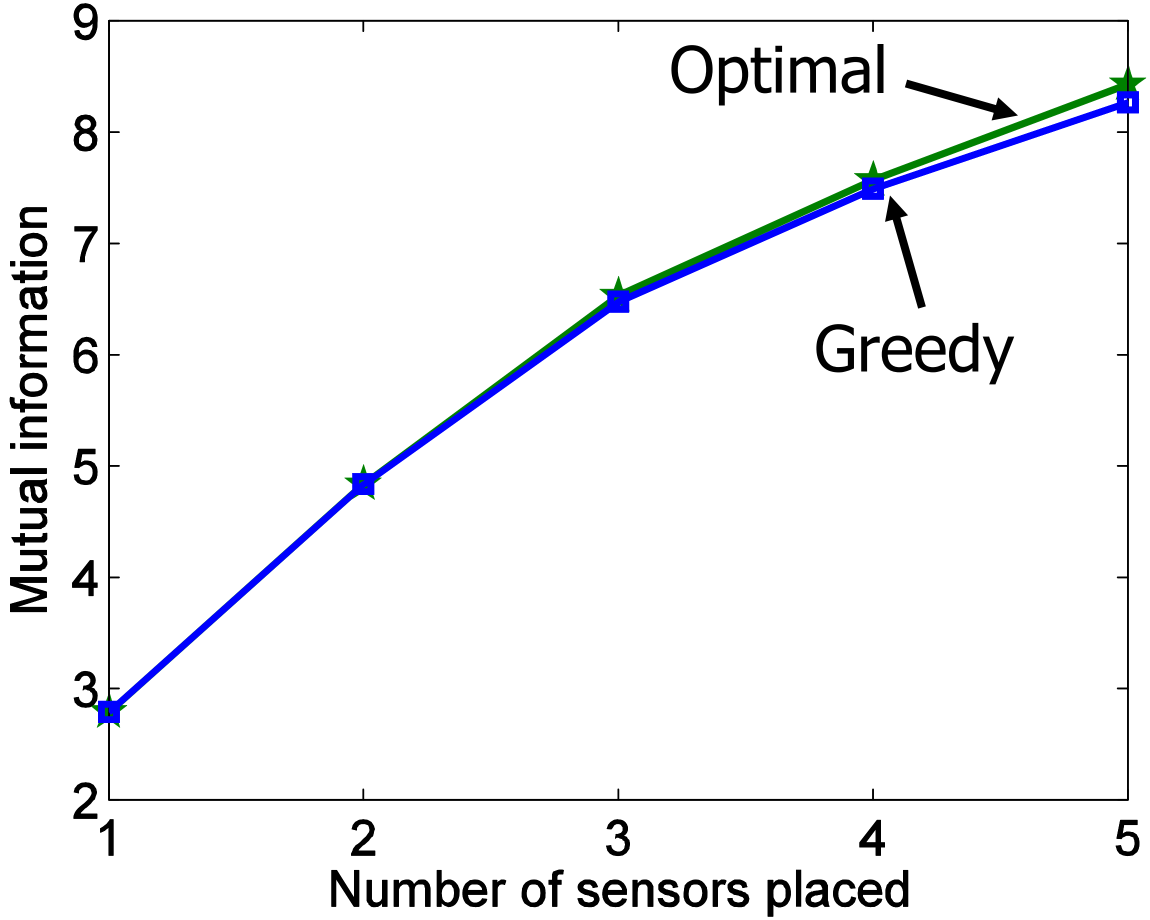
For i = 1 to k

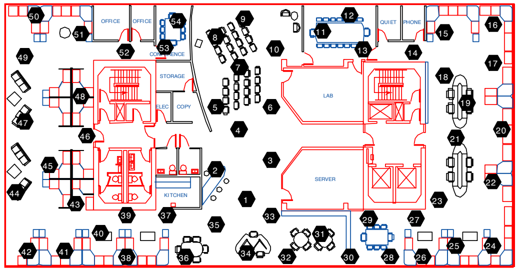
s* := argmax$_s$ F(A [{s})

A := A [{s*}

How well can this simple heuristic do?

Performance of greedy



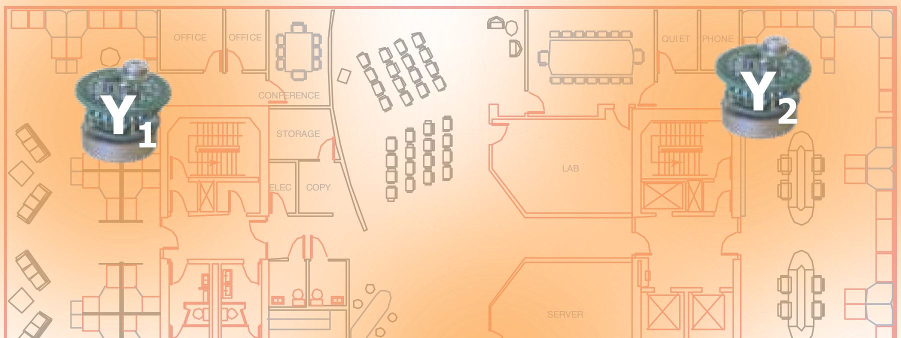


Temperature data from sensor network

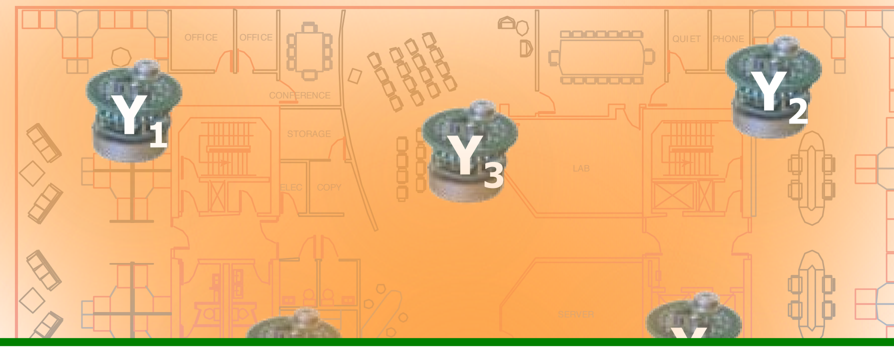
- Greedy empirically close to optimal. Why?

Key observation: Diminishing returns

Placement A = {Y_1, Y_2}



Placement B = {Y_1,…, Y_5}

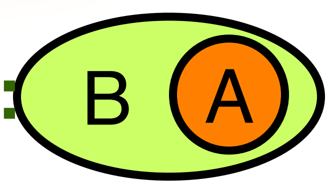


Theorem [Krause and Guestrin, UAI '05]**:**

Information gain $F(A) = H(X) – H(X \mid Y_A)$ is submodular!

New sensor Y'

Submodularity:



+ •Y' Large improvement

+ •Y' Small improvement

For A µ B, F(A [{Y'}) – F(A) ¸ F(B [{Y'}) – F(B)

One reason submodularity is useful

Theorem [Nemhauser et al '78]

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!
- For information gain: Guarantees best possible unless P = NP! [Krause & Guestrin '05]

Challenges for environmental monitoring

Use robots to monitor environment

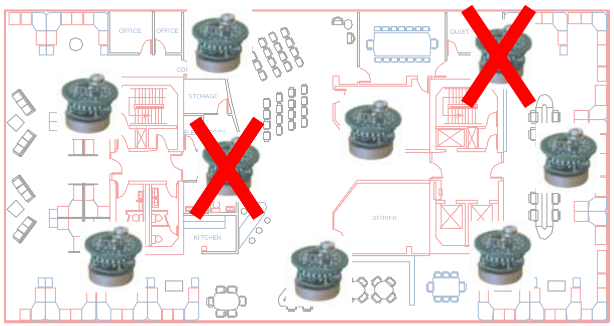
Not just select best k locations A for given F(A). Need to

… be robust against uncertainty in the function F

… take into account cost of traveling between locations

… cope with environments that change over time

Why do we need robustness?



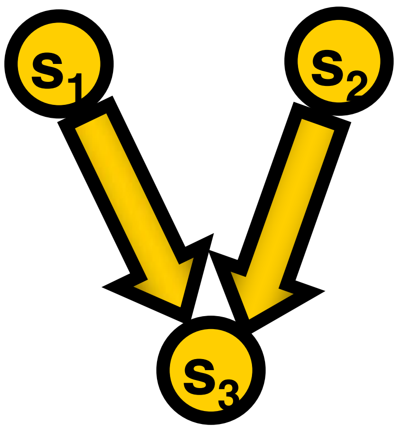
→ Sensor failures

→ Adversarial environments

Unified view: $\mathcal{A}^* = \underset{|\mathcal{A}|\leq k}{\operatorname{argmax}} \min_i F_i(\mathcal{A})$

How does the greedy algorithm do?

$V=\{s_1, s_2, s_3\}$ Buy k=2 sensors F_i = intrusion at s_i



Optimal solution

Optimal score: 1

Greedy picks s_3 first

Then, can choose only s_1 or s_2

Greedy score: ε

➔ Greedy does arbitrarily badly ☹
Can we do better?