Introduction to Artificial Intelligence

Lecture 14 – Information gathering

CS/CNS/EE 154

Andreas Krause

TexPoint fonts used in EMF.

Announcements

- Homework 2 due today
- Homework 3 out later this week
- Final project due December 1
- Code released on Monday (Nov 15)

- Note on midterm grades (Avian Asker)

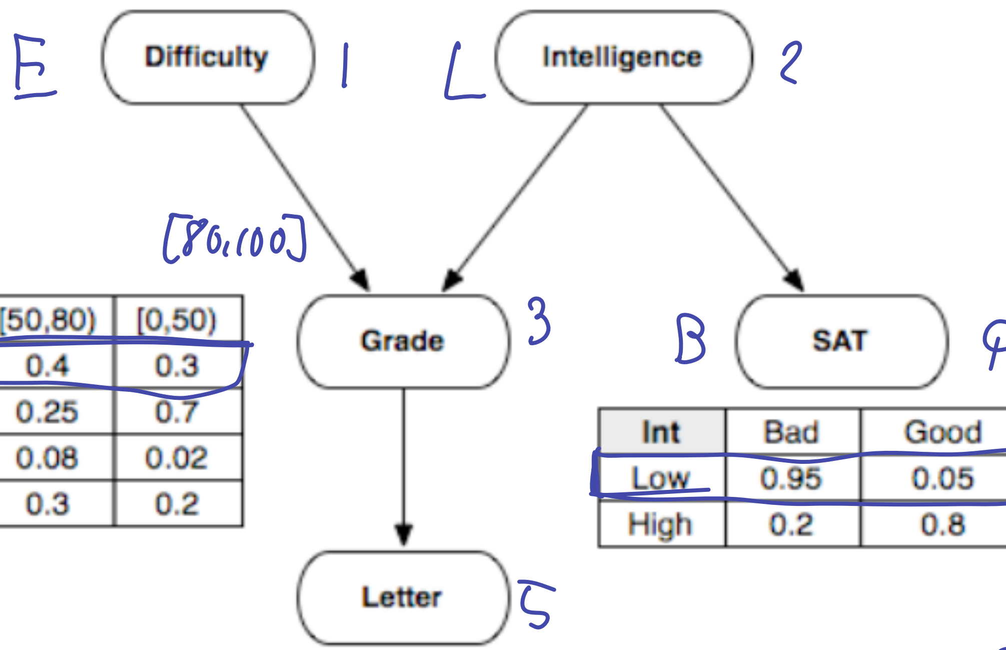
Sampling based inference

- So far: deterministic inference techniques
  - Variable elimination
  - (Loopy) belief propagation
- Will now introduce stochastic approximations
  - Algorithms that “randomize” to compute expectations
  - In contrast to the deterministic methods, guaranteed to converge to right answer (if wait looong enough..)
  - More exact, but slower than deterministic variants

Forward sampling from a BN

Easy	Hard
0.6	0.4

Low	High
0.7	0.3



Int	Diff	[80,100]	[50,80)	[0,50)
Low	Easy	0.3	0.4	0.3
Low	Hard	0.05	0.25	0.7
High	Easy	0.9	0.08	0.02
High	Hard	0.5	0.3	0.2

Int	Bad	Good
Low	0.95	0.05
High	0.2	0.8

P

Grade	Fail	Pass
[80,100]	0.1	0.9
[50,80)	0.4	0.6
[0,50)	0.99	0.01

Sample

$\Rightarrow$ (D=E, I=L, G=[80,100], S=B, L=P)

Rejection sampling

- Collect samples over all variables

$$\widehat{P}(\mathbf{X}_A = \mathbf{x}_A \mid \mathbf{X}_B = \mathbf{x}_B) \approx \frac{Count(\mathbf{x}_A, \mathbf{x}_B)}{Count(\mathbf{x}_B)}$$

- Throw away samples that disagree with $\mathbf{x_B}$
- Can be problematic if P($\mathbf{X_B}$ = $\mathbf{x_B}$) is rare event

Sample complexity for probability estimates

- Absolute error:

$$Prob\Big(\big|\widehat{P}(\mathbf{x}) - P(\mathbf{x})\big| > \varepsilon\Big) \leq 2\exp(-2N\varepsilon^2)$$

- Relative error:

$$Prob\Big(\widehat{P}(\mathbf{x}) < (1+\varepsilon)P(\mathbf{x})\Big) \leq 2\exp(-NP(\mathbf{x})\varepsilon^2/3)$$

if $P(x)$ exponentially small, need N exponentially large

Sampling from rare events

- Estimating conditional probabilities $P(X_A \mid \mathbf{X_B}=\mathbf{x_B})$ using rejection sampling is hard!
  - The more observations, the unlikelier $P(\mathbf{X_B} = \mathbf{x_B})$ becomes

- Want to directly sample from posterior distribution!

Gibbs sampling

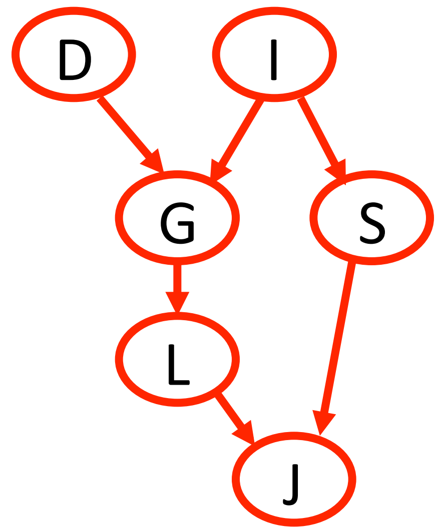
- Start with initial assignment $\mathbf{x}^{(0)}$ to all variables
- For t = 1 to ∞ do
  - Set $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$
  - For each variable X_i
    - Set $\mathbf{v_i}$ = values of all $\mathbf{x}^{(t)}$ except x_i
    - Sample $x^{(t)}_i$ from $P(X_i \mid \mathbf{v_i})$

- For large enough *t*, sampling distribution will be "close" to true posterior distribution!
- **Key challenge**: Computing conditional distributions $P(X_i \mid \mathbf{v_i})$

Gibbs Sampling

Gibbs sampling P(D,I,G,S,L | J = 1)

Iter	D	I	G	S	L	J
1	0	1	1	0	0	**1**
2	* 0	1	0	0	1	**1**
3	0	0	1	0	0	**1**
4	0	0	1	0	1	**1**
…						



$P(G=1) \approx \frac{3}{4}$

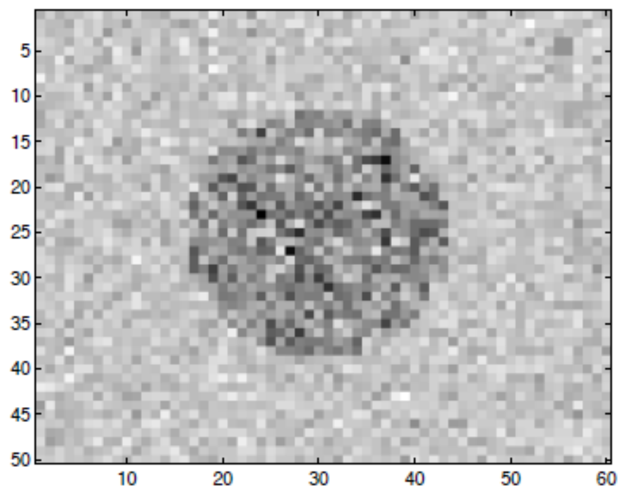
$* \sim P(D \mid I=1, G=1, S=0, L=0, J=1)$

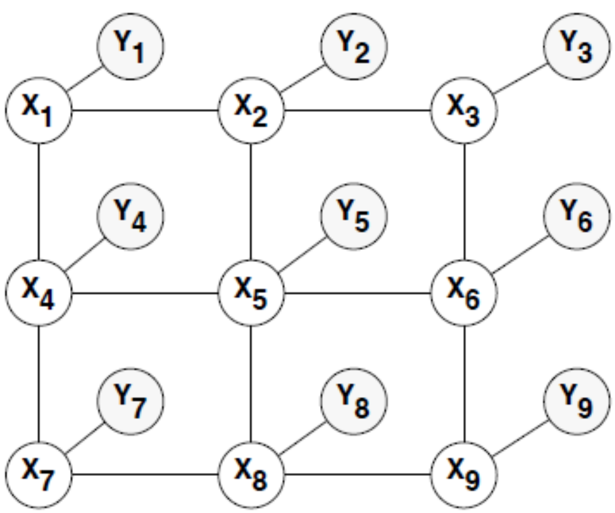
$= \frac{1}{Z} P(D, I=1, G=1, S=0, L=0, J=1) =$

$$= \frac{P(D)\,\cancel{P(I=1)}\,P(G=1 \mid D, I=1)\,\cancel{P(S=0 \mid I=1)} \cdot \cancel{P(L=0 \mid G=1)}\,\cancel{P(J=1 \mid L=0, S=0)}}{\sum_d P(D=d)\,\cancel{P(I=1)}\,P(G=1 \mid D=d, I=1)\,\cancel{P(S=0 \mid I=1)}\,\cancel{P(L=0 \mid G=1)}\,\cancel{P(J=1 \mid L=0, S=0)}}$$

$= \frac{1}{Z'} P(D)\,P(G=1 \mid D, I=1)$

Example: (Simple) image segmentation





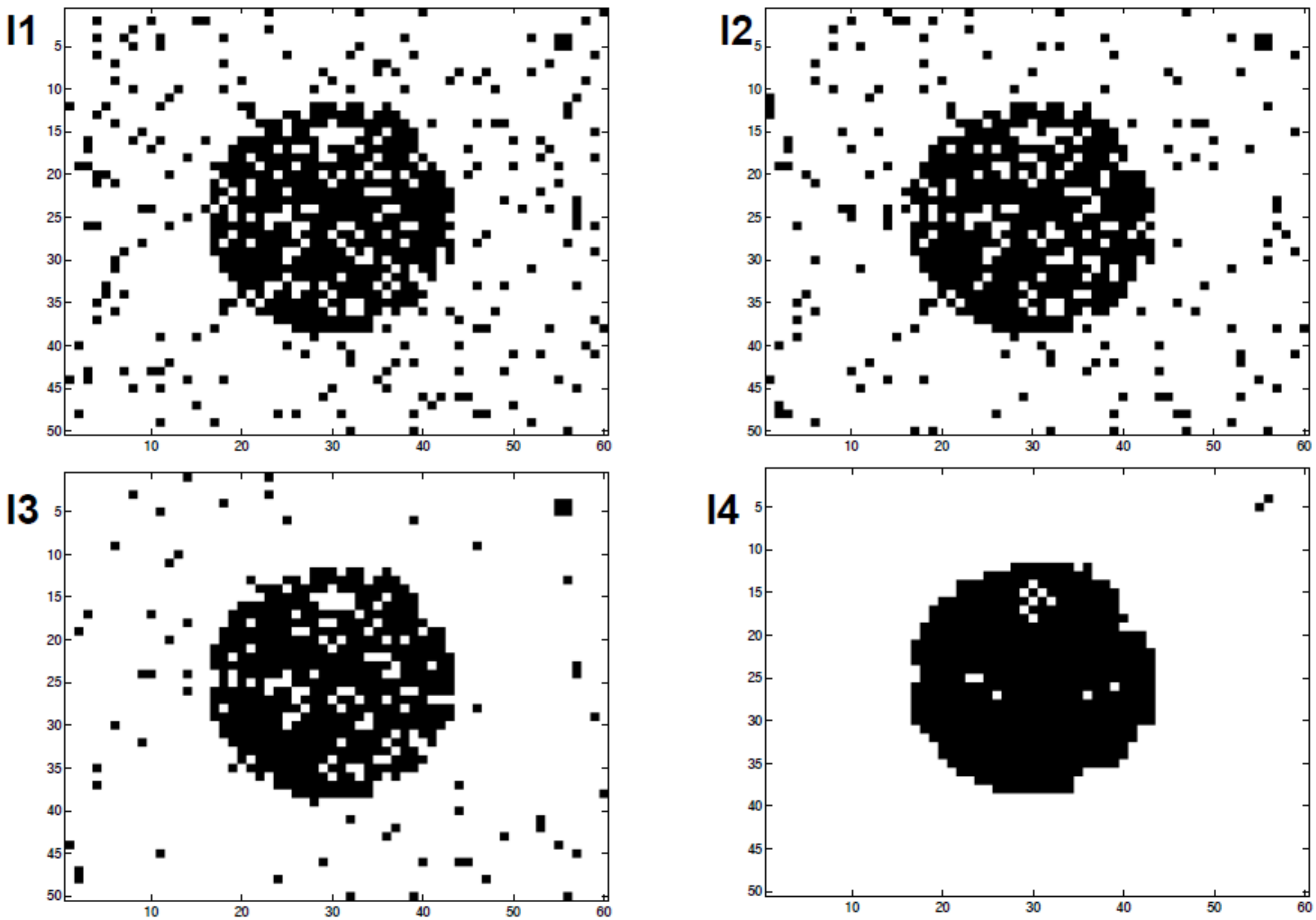
$$P(x) = \frac{1}{Z} \prod_i \Phi(x_i) \prod_{(j,k)\in E} \Psi(x_j, x_k)$$

$$\Phi(x_i) = exp\left\{ -\frac{(y_i - \mu_{x_i})^2}{2\sigma_{x_i}^2} \right\}$$

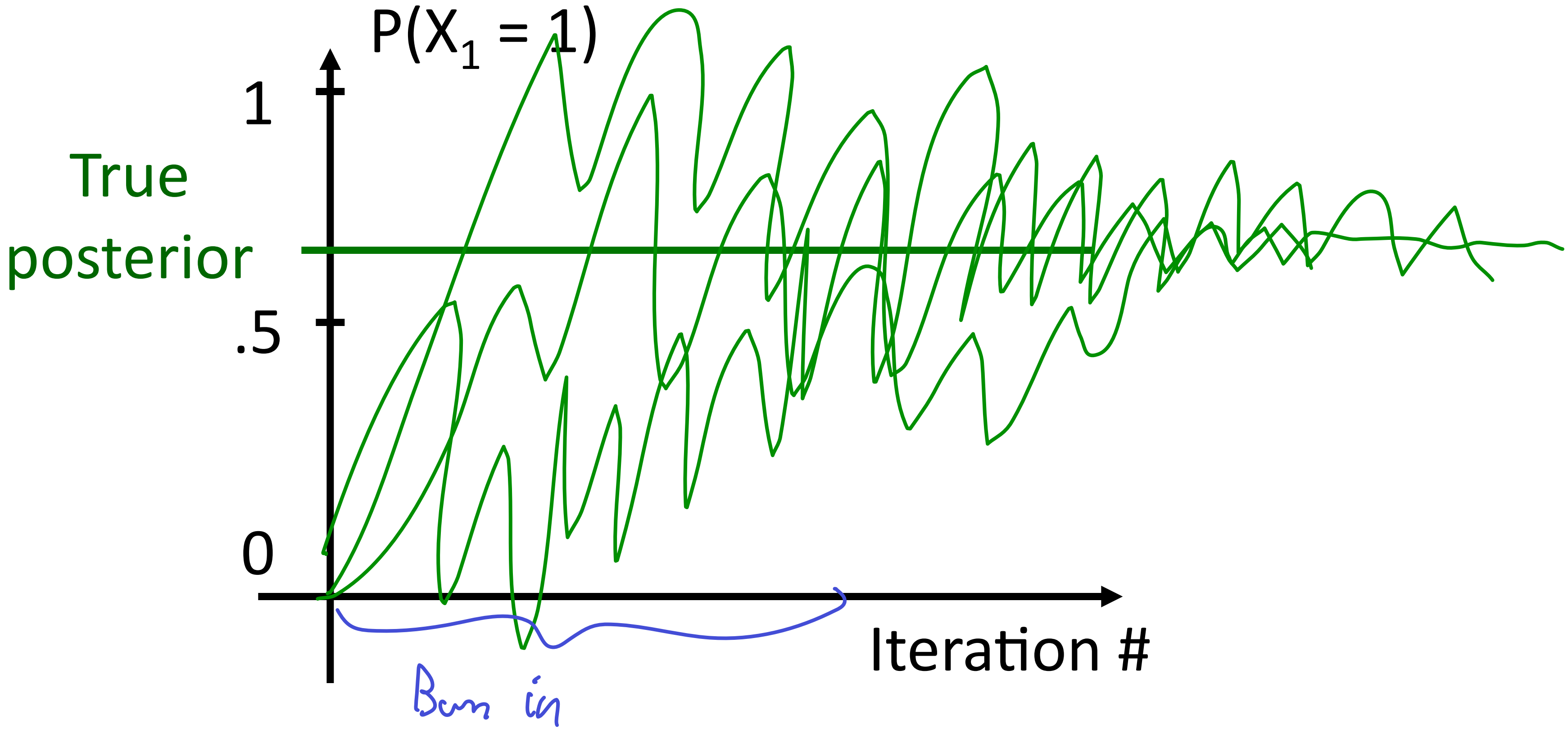
$$\Psi(x_i, x_j) = \exp\left\{ -\beta(x_i - x_j)^2 \right\}$$

[see Singh '08]

Gibbs Sampling iterations



Convergence of Gibbs Sampling



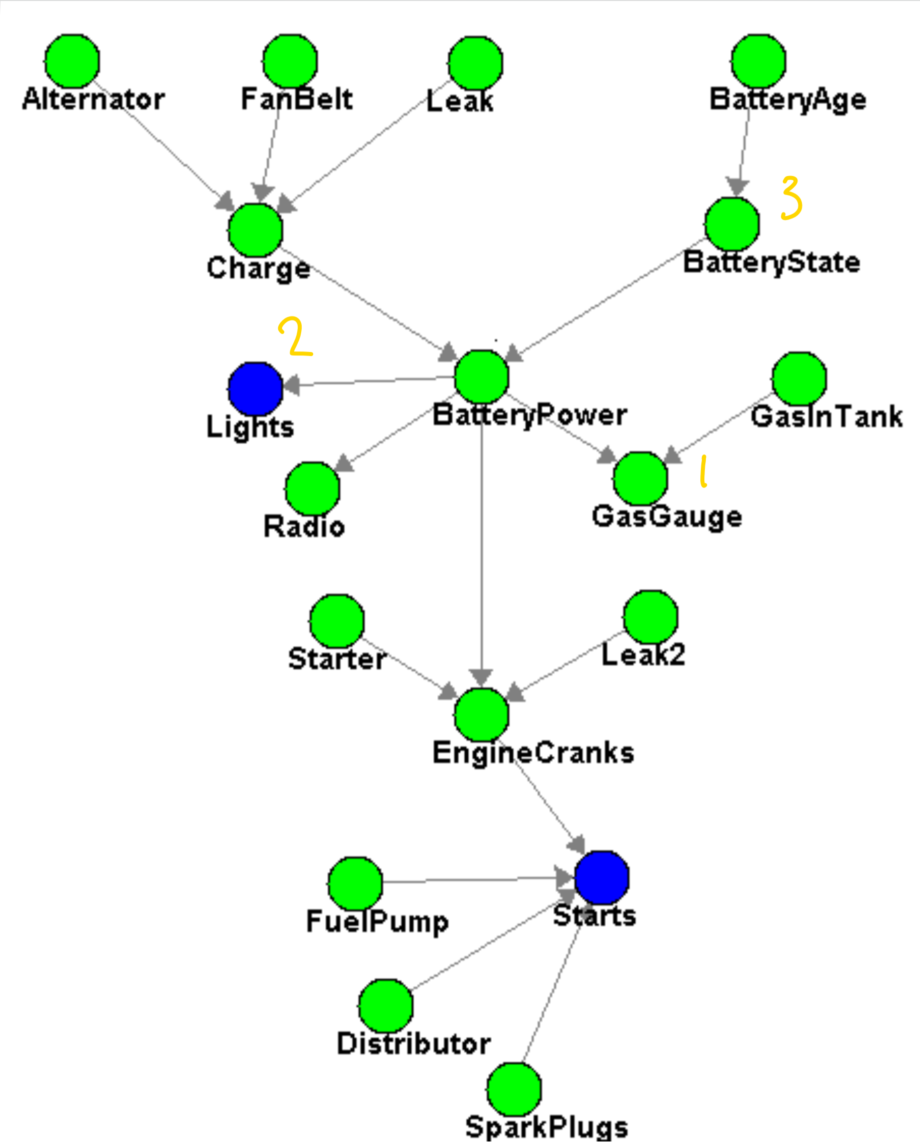
Summary: Inference

- For tree-structured Bayes nets, can compute exact marginals
  - Variable elimination
  - Belief propagation (efficiently computes all marginals)
- For loopy networks, can use approximate inference
  - Loopy belief propagation (may not converge)
  - Gibbs sampling (will converge, but may take long time)

Information gathering

- So far:
  - Bayesian networks for quantifying uncertainty in real world environments
  - Exact and approximate algorithms for inference in Bayesian networks (e.g., compute P(Pit | Breezes))
- Now:
  - Selecting most "informative" variables for making effective predictions / decisions

Why does my car not start?



- Selectively run tests to diagnose cause of failure

Clinical diagnosis?

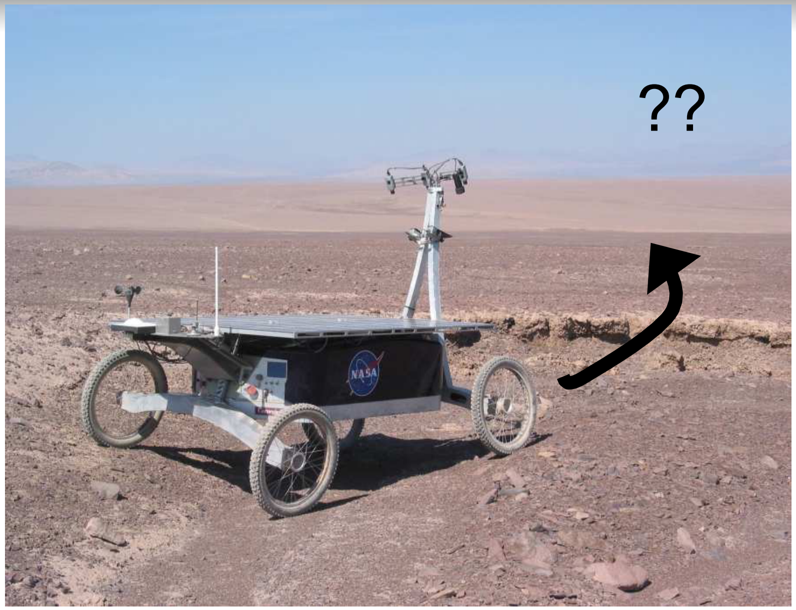
- Patient either healthy or ill
- Can choose to treat or not treat

	healthy	*ill*
Treatment	-$$	$
No treatment	0	-$$$

- Only know distribution P(ill | observations)
- Can perform costly medical tests to reveal aspects of the condition

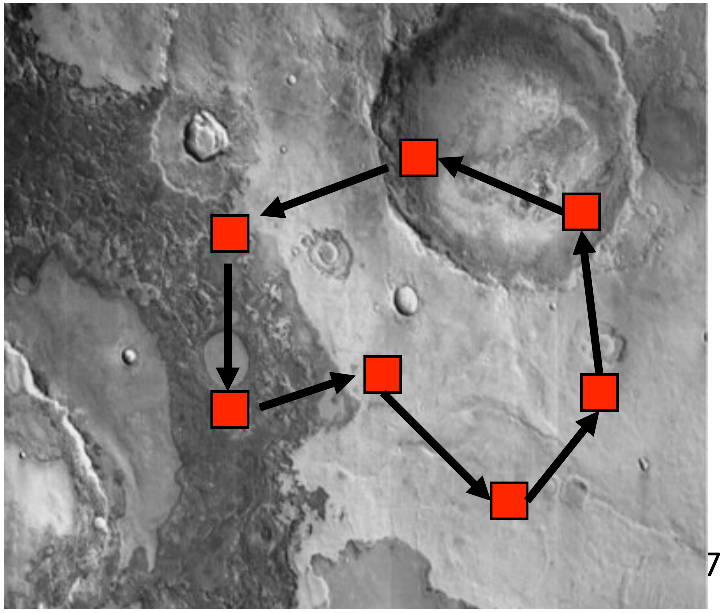
- **Which tests should we perform to most cost-effectively diagnose?**

Autonomous robotic exploration

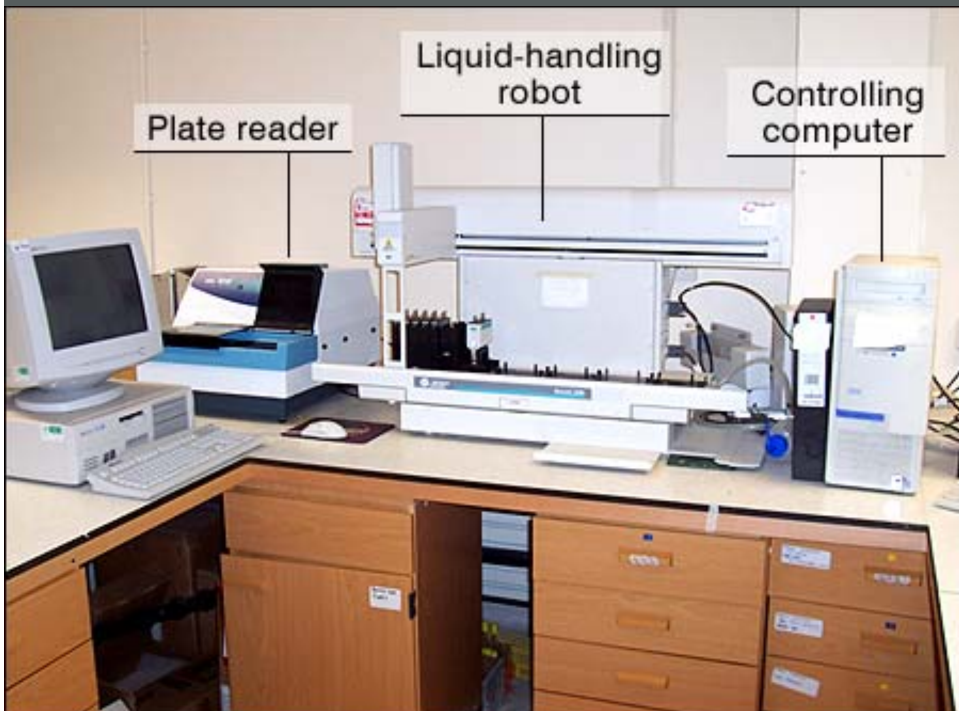


- Limited time for measurements
- Limited capacity for rock samples

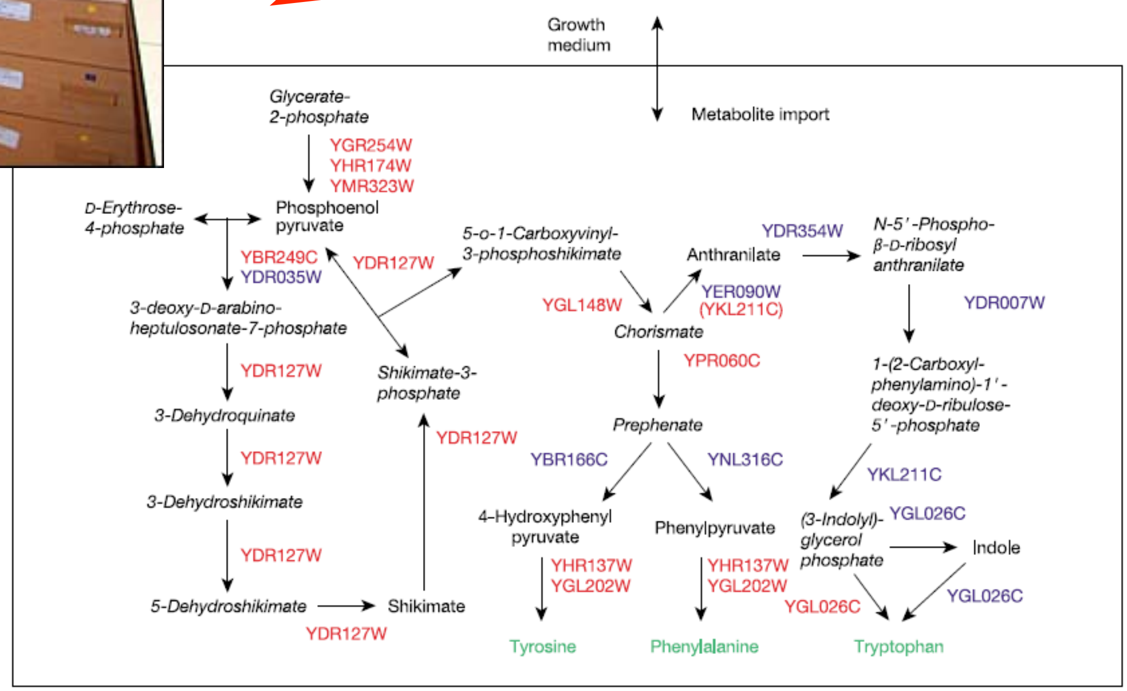
- **Need optimized information gathering!**



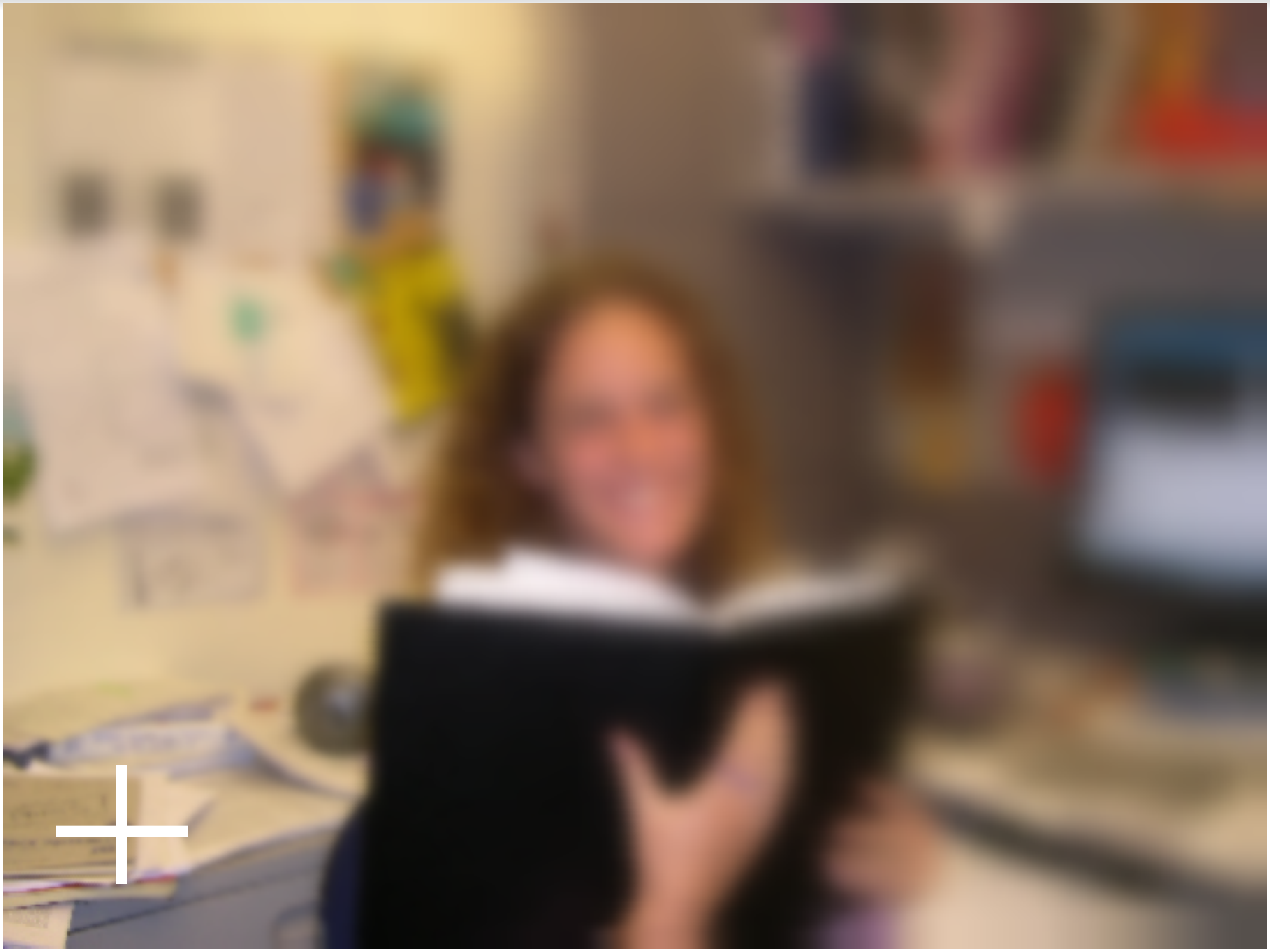
A robot scientist



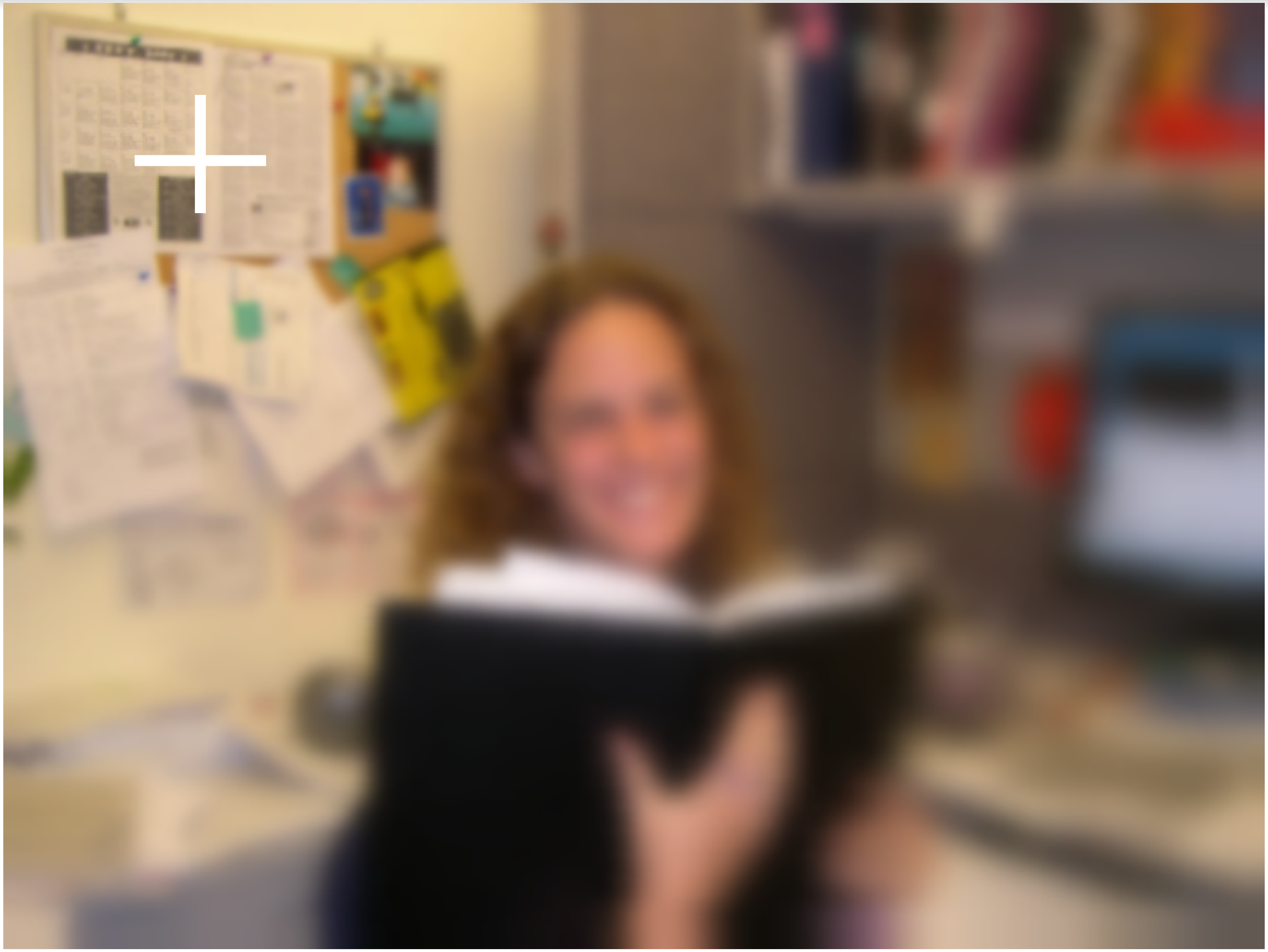
King et al, Nature '04



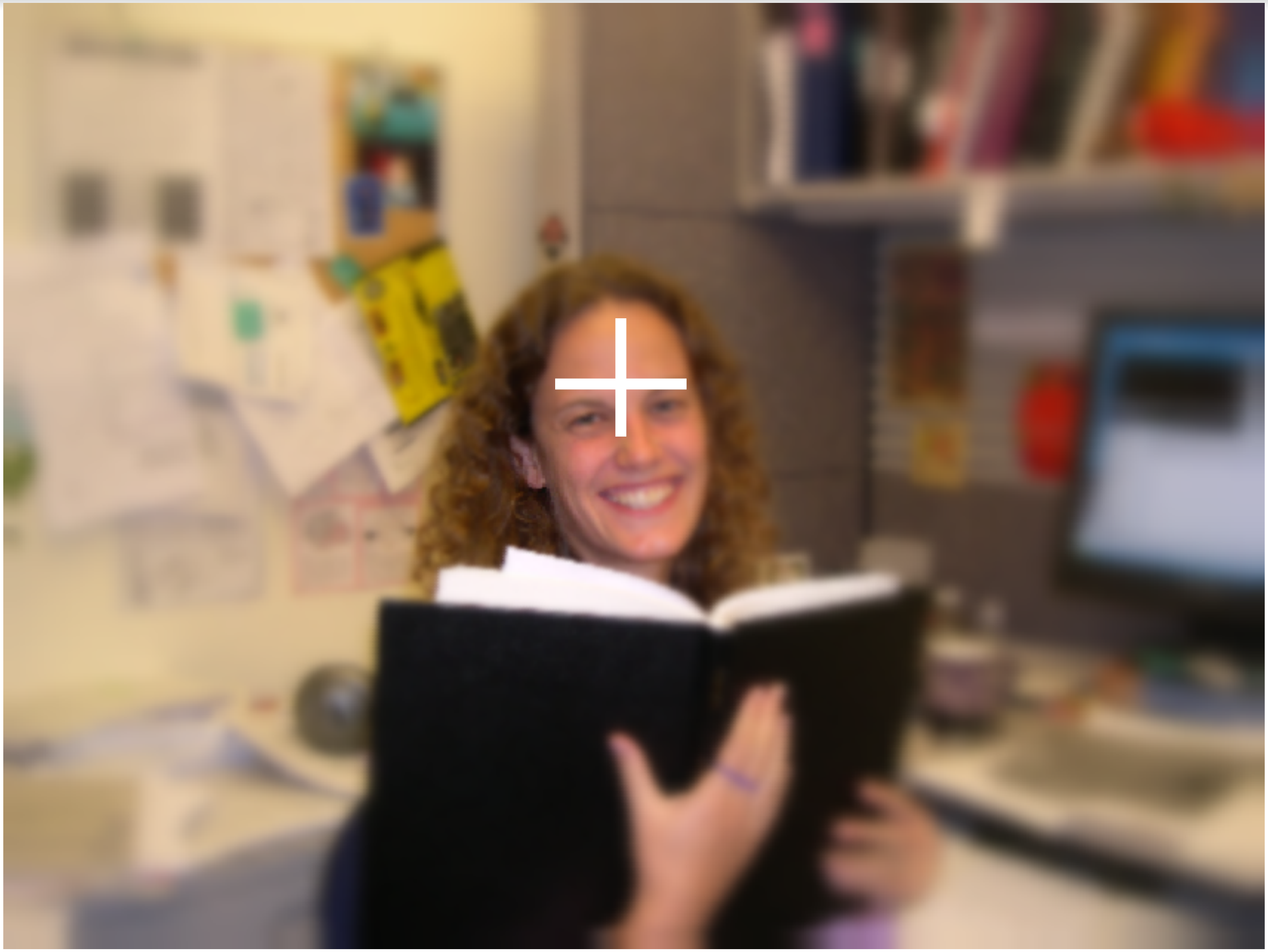
How do people gather information?
[Renninger et al, NIPS ’04]



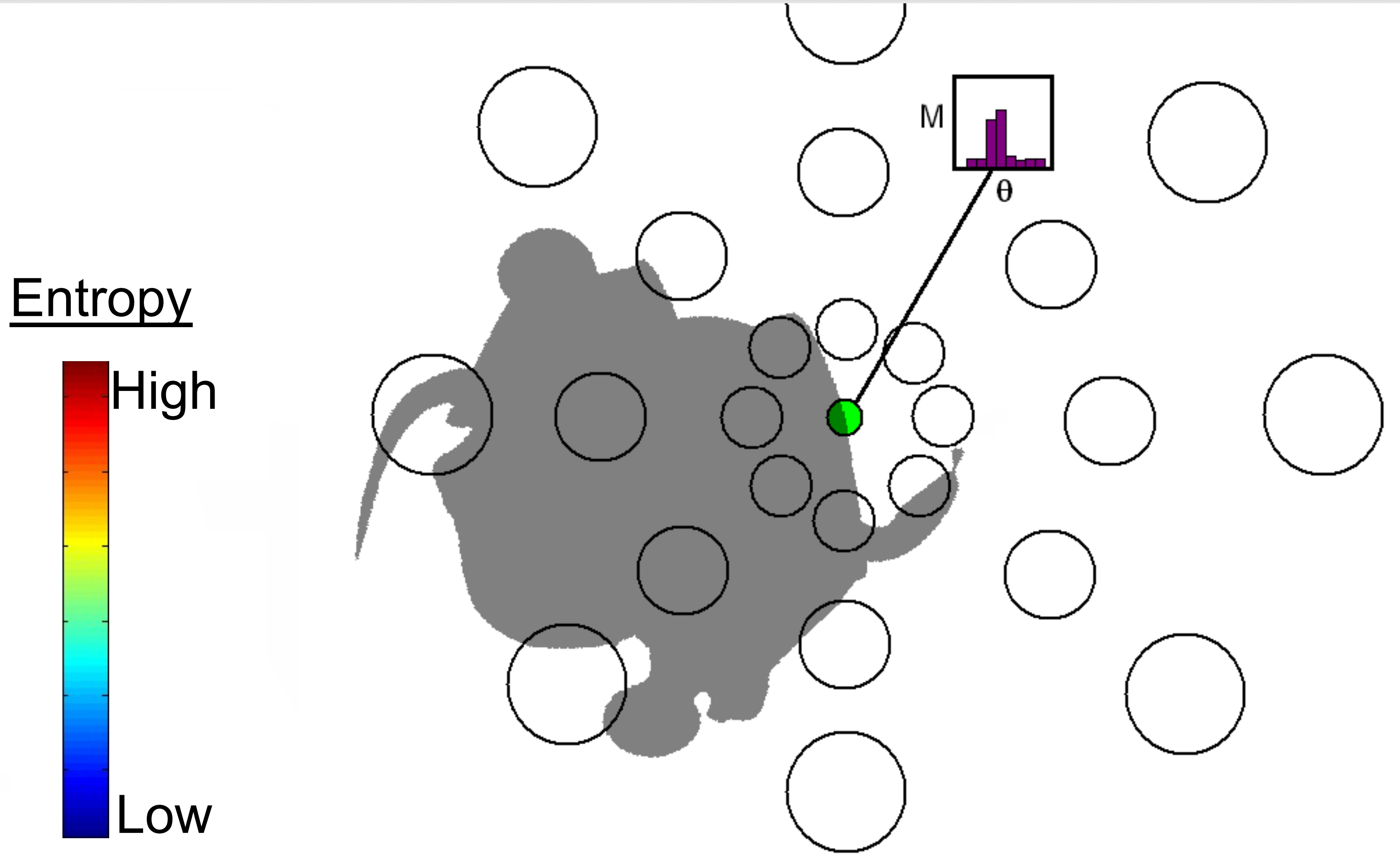
How do people gather information?
[Renninger et al, NIPS '04]



How do people gather information?
[Renninger et al, NIPS '04]

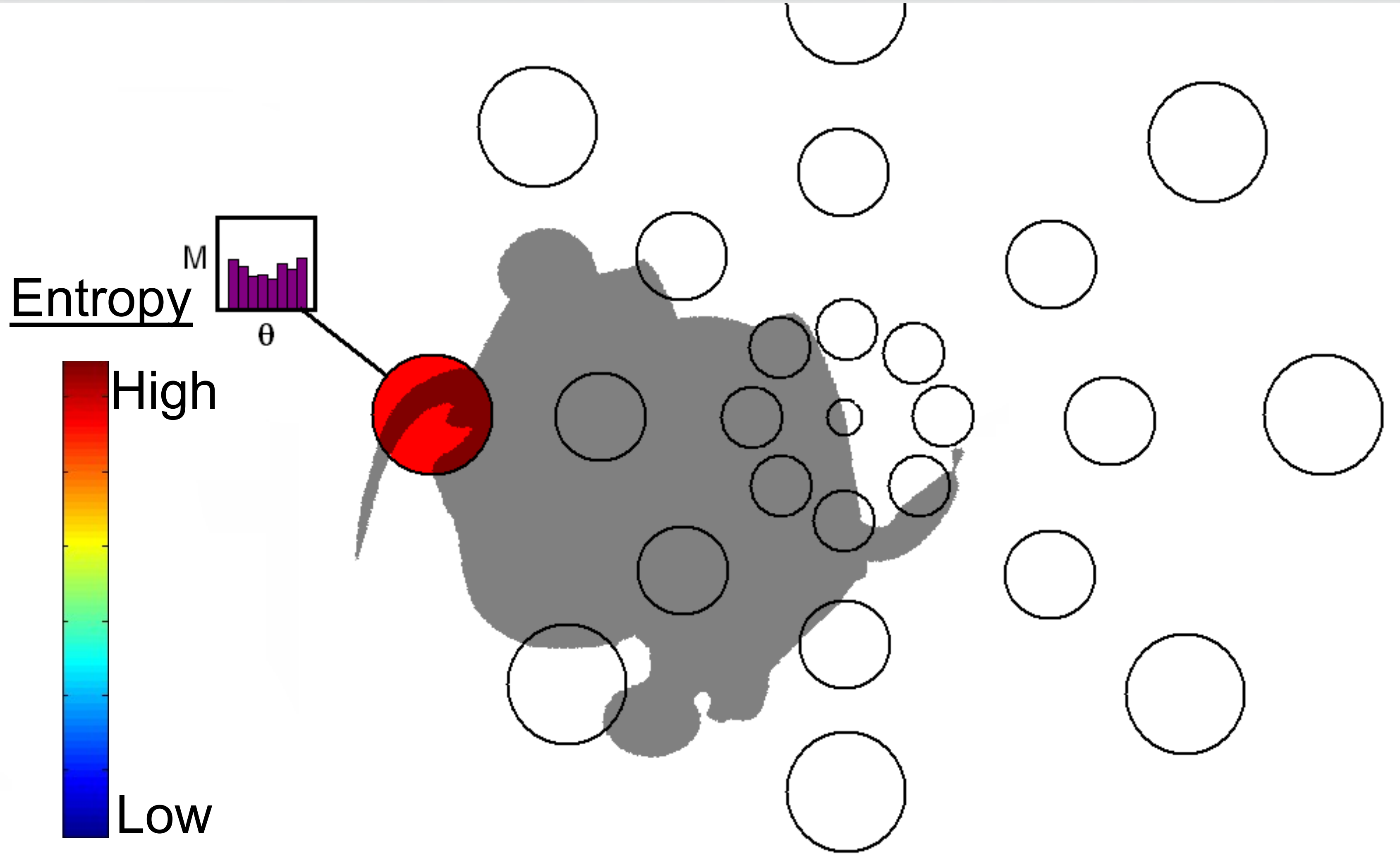


How do people gather information?
[Renninger et al, NIPS '04]



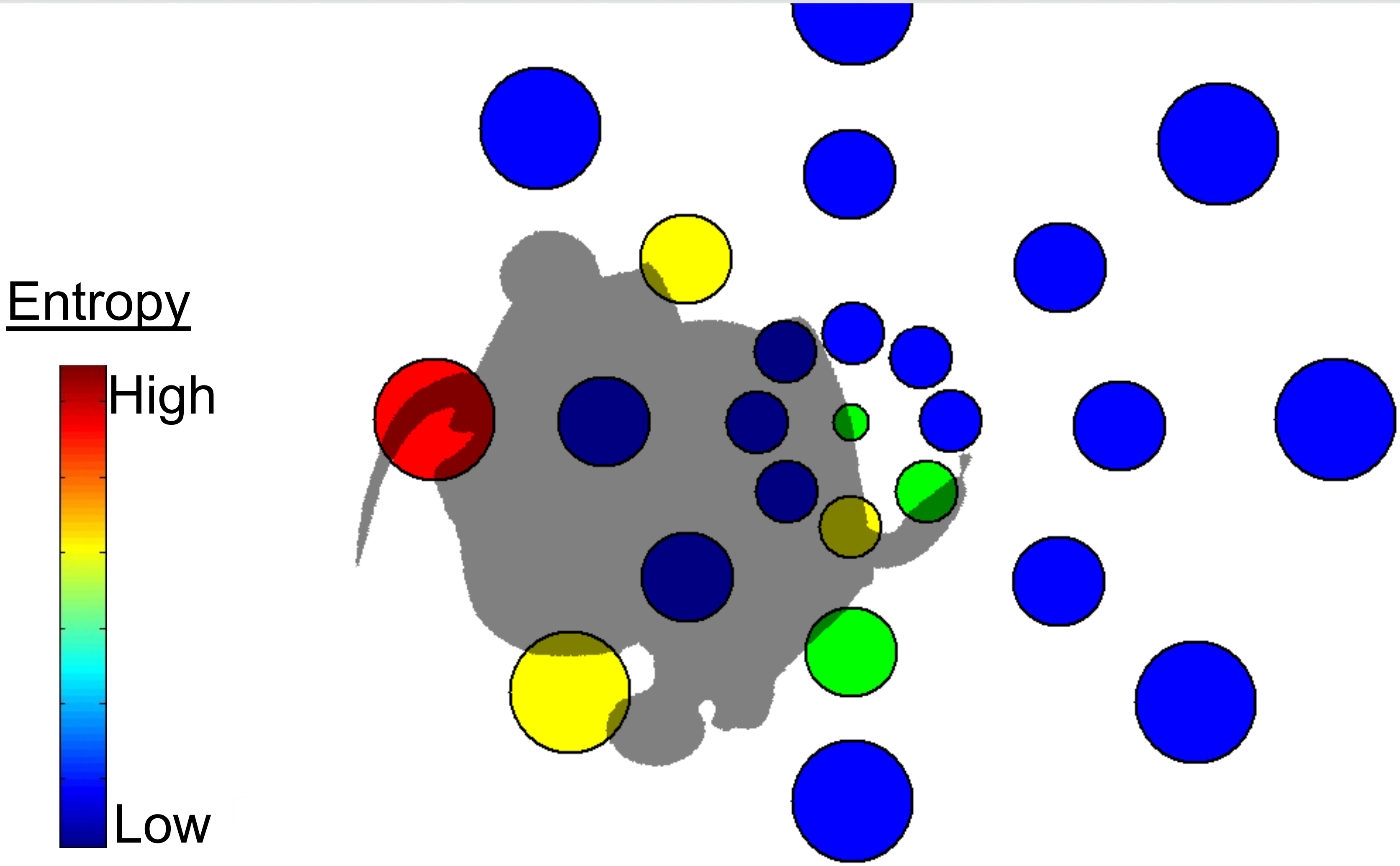
How do people gather information?

[Renninger et al, NIPS ’04]

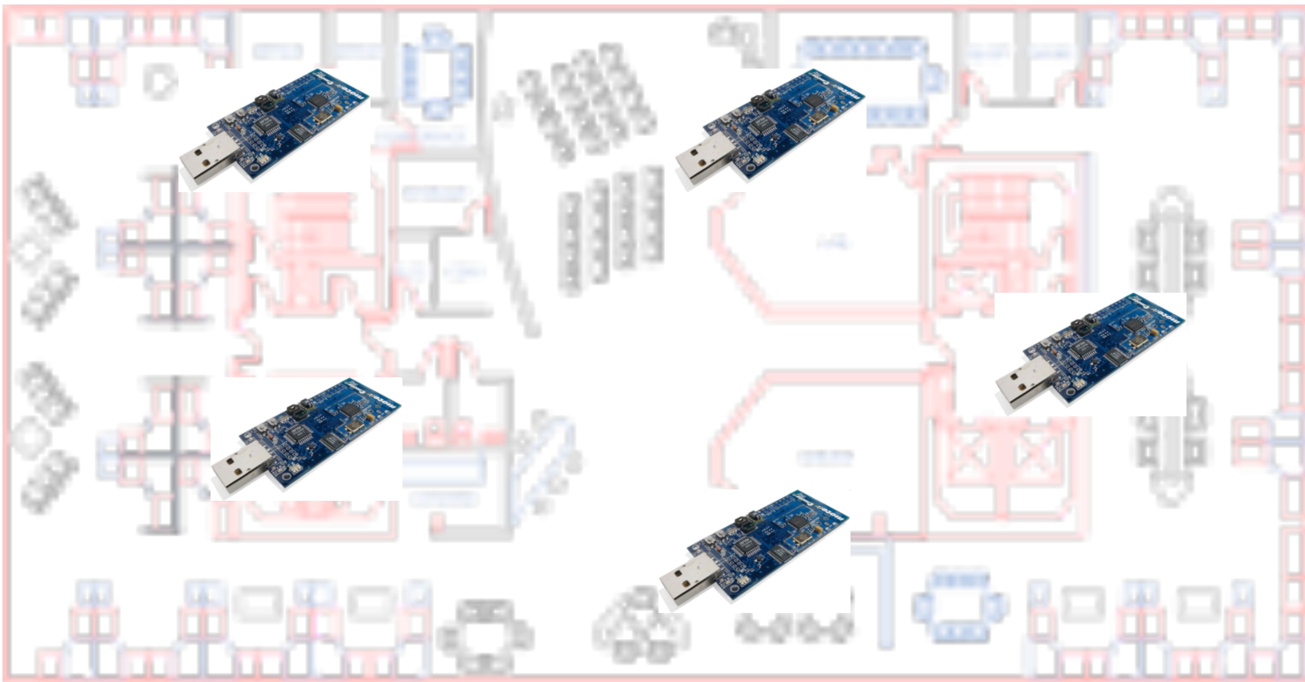


How do people gather information?

[Renninger et al, NIPS '04]

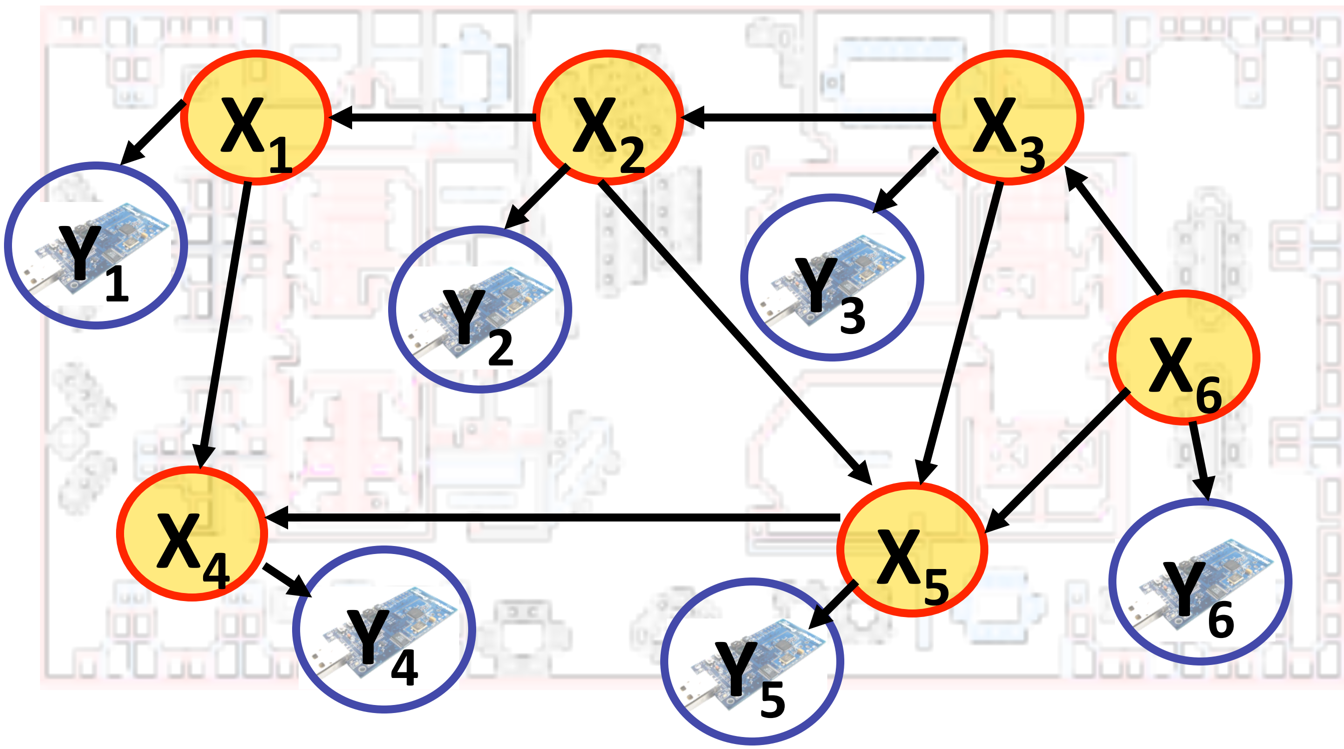


Running example: Detecting fires



Want to place sensors to detect fires in buildings

Monitoring using Bayesian Networks



X_s: temperature at location s

Y_s: sensor value at location s

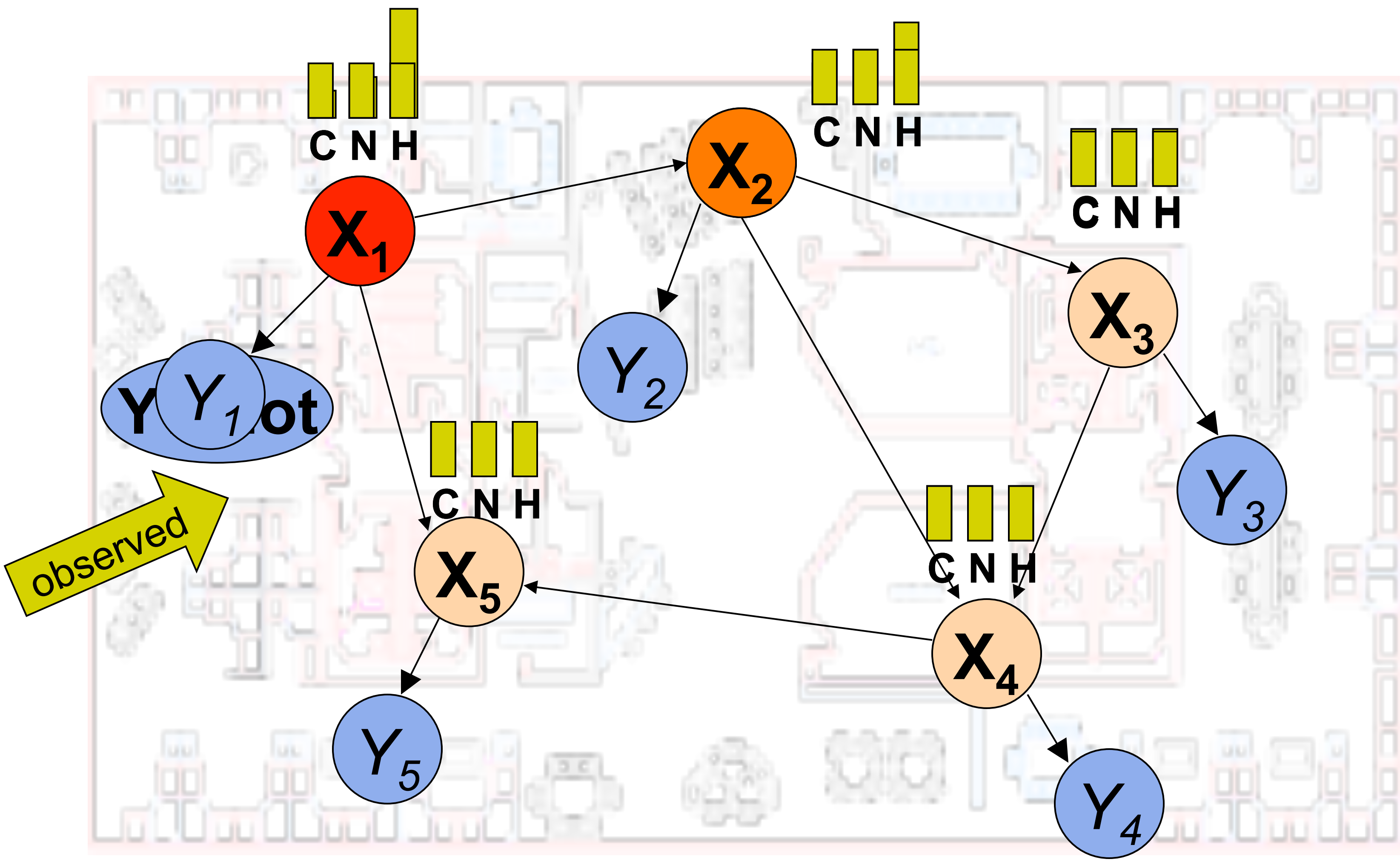
$$Y_s = X_s + \text{noise}$$

Joint probability distribution

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

26

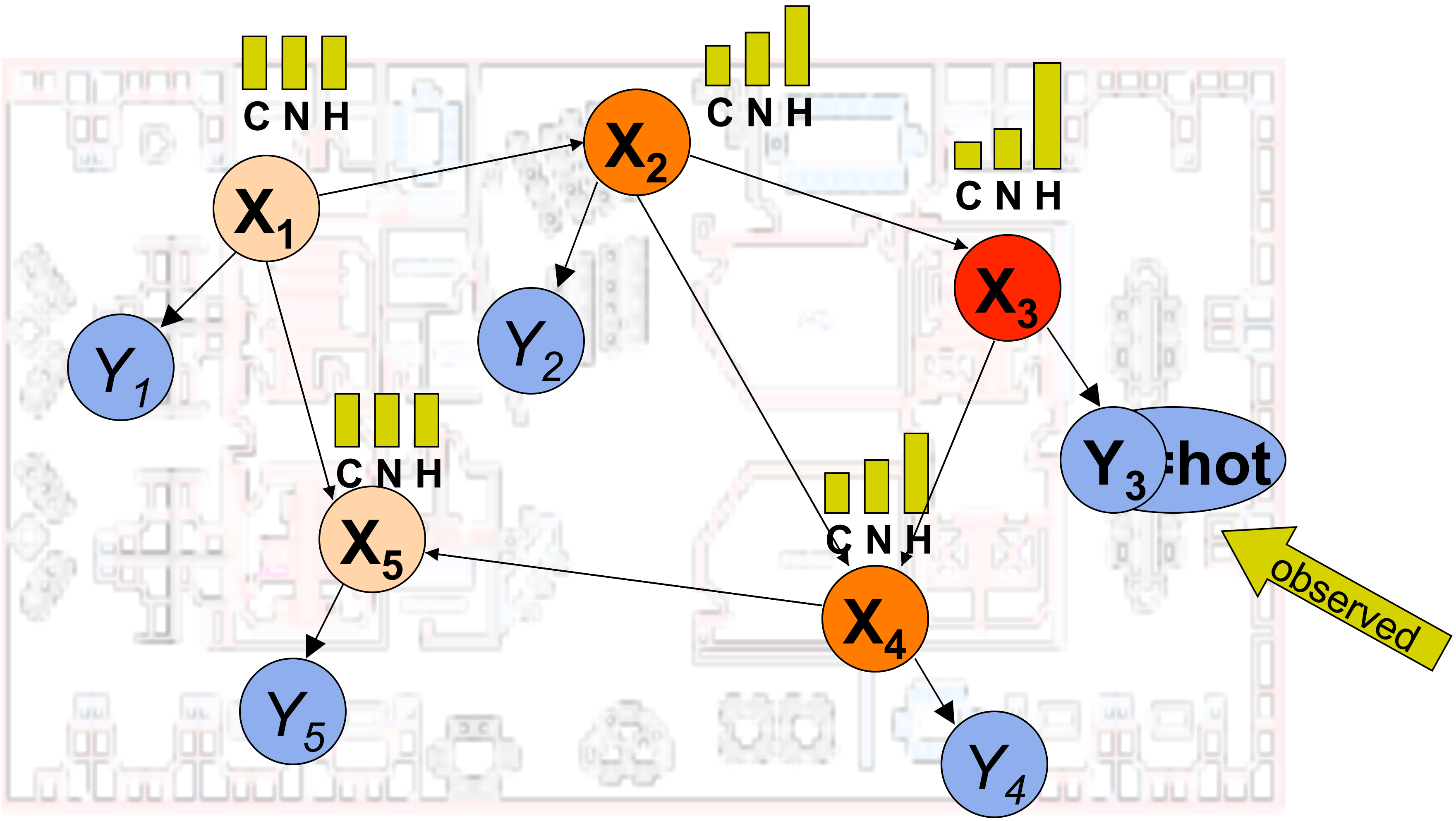
Making observations



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

27

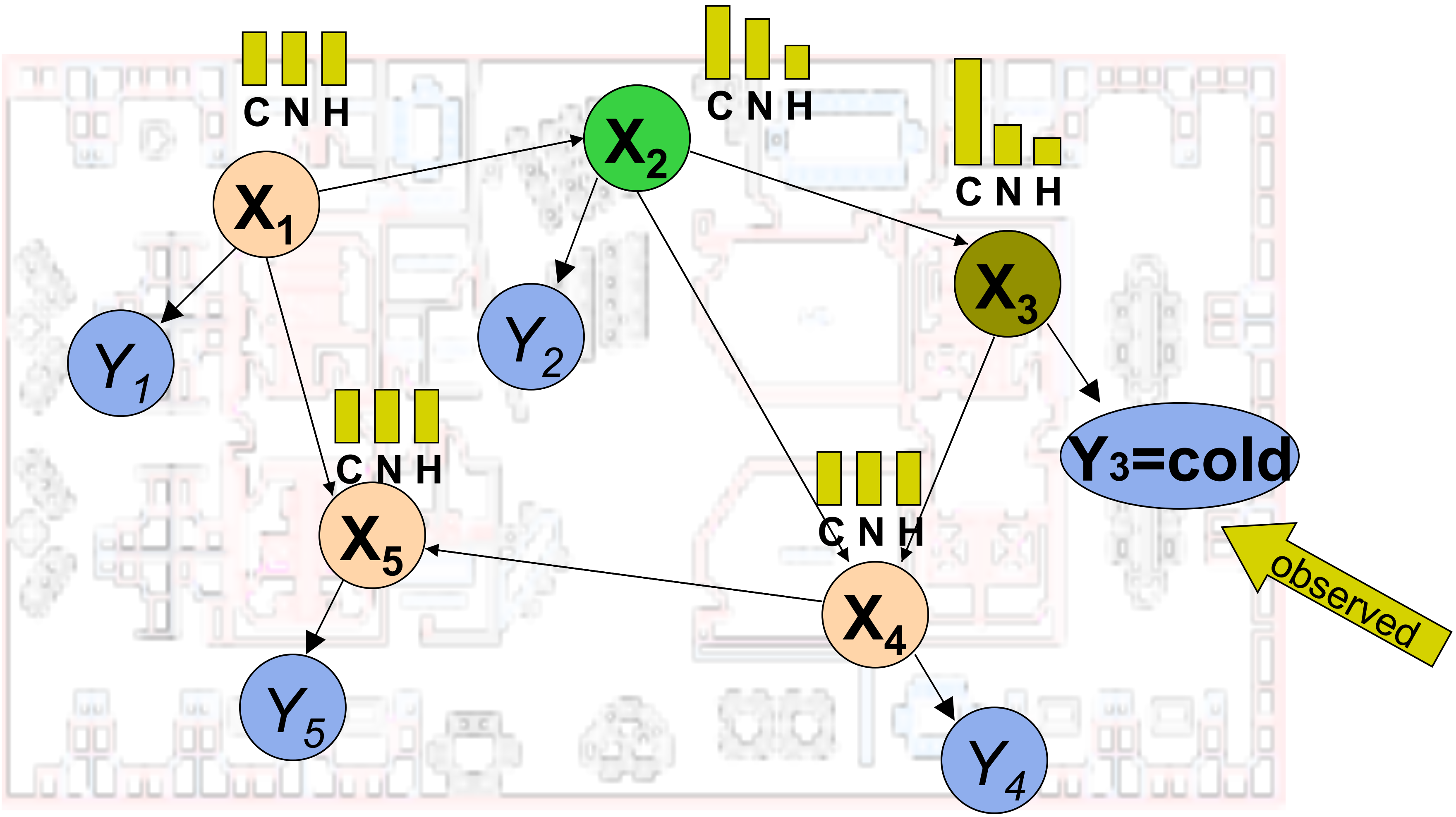
Making observations



Reward[P(**X**|Y_3=hot)] = 0.4

28

A different outcome…



Reward[P($\mathbf{X}$|Y_3=cold)] = 0.1

29

Reducing uncertainty

- Want to select observations that maximize reduction in uncertainty

- Can quantify uncertainty using Shannon entropy:

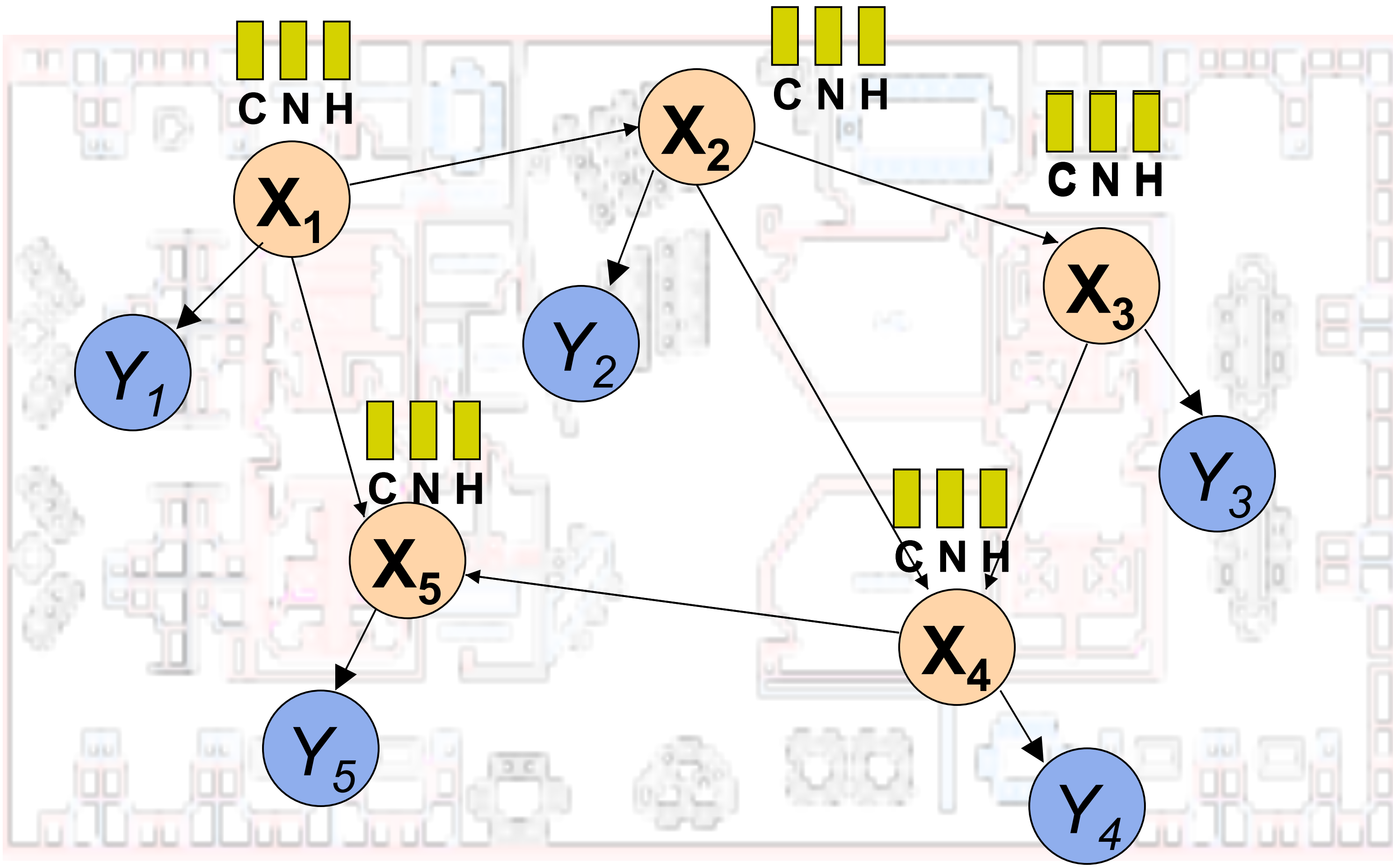
$$H(X) = -\sum_x P(X = x) \log_2 P(X = x)$$

- For discrete variables $0 \leq H(X) \leq \log_2 |dom(X)|$

Sps. $P(X=x) = \frac{1}{n}$ $\Rightarrow H(X) = -n \cdot \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n$

- Thus, can use Reward[P(**X**)] = -H(**X**) = $\sum_x$ P(**x**) $\log_2$ P(**x**)

Making observations



Prior entropy: $H(\mathbf{X}) \approx 4.2$

31

Posterior entropy

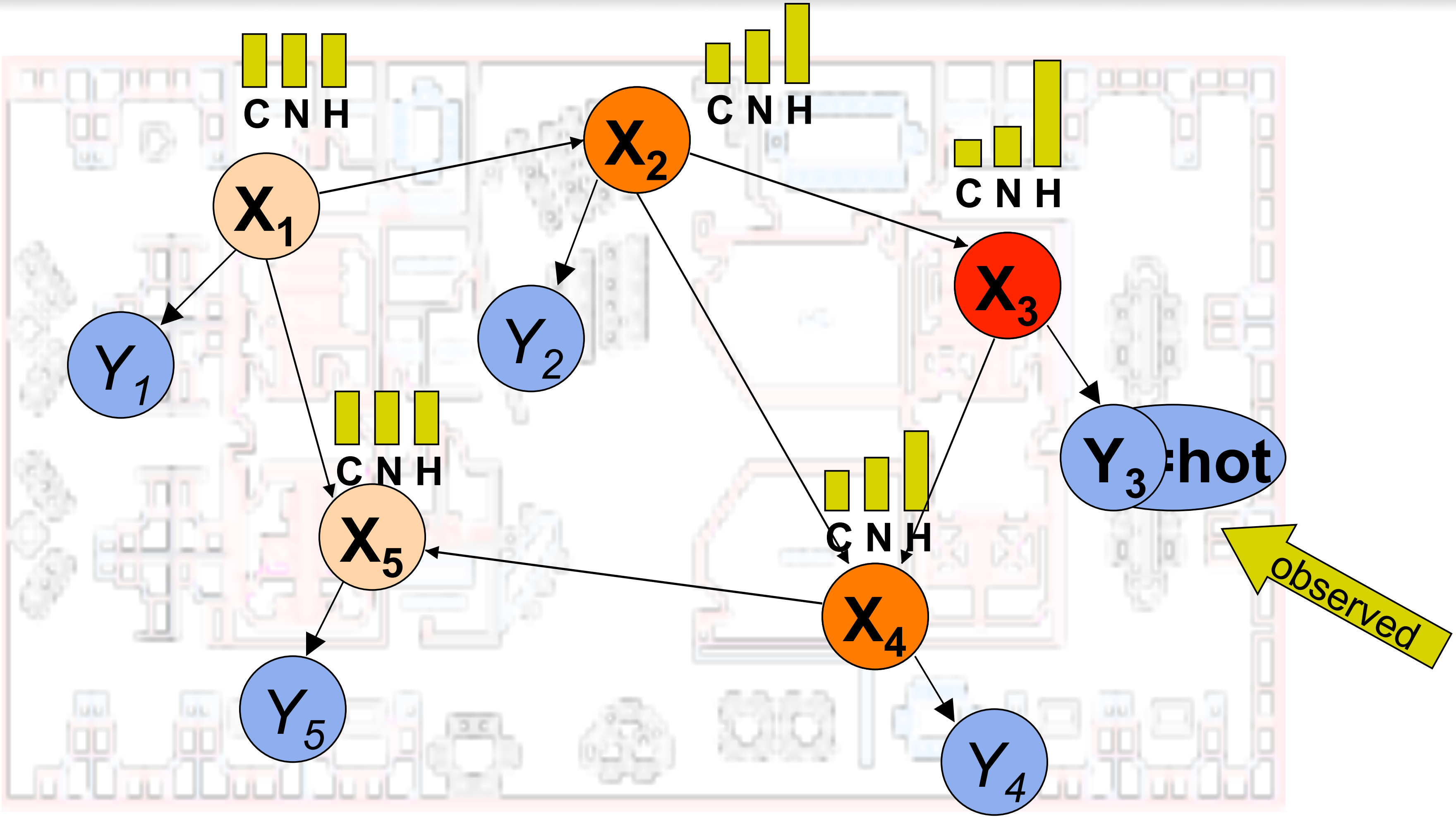
- Entropy before observations:

$$H(X) = -\sum_x P(X = x) \log_2 P(X = x)$$

- Entropy *after* observing Y = y:

$$H(X \mid Y = y) = -\sum_x P(X = x \mid Y = y) \log_2 P(X = x \mid Y = y)$$

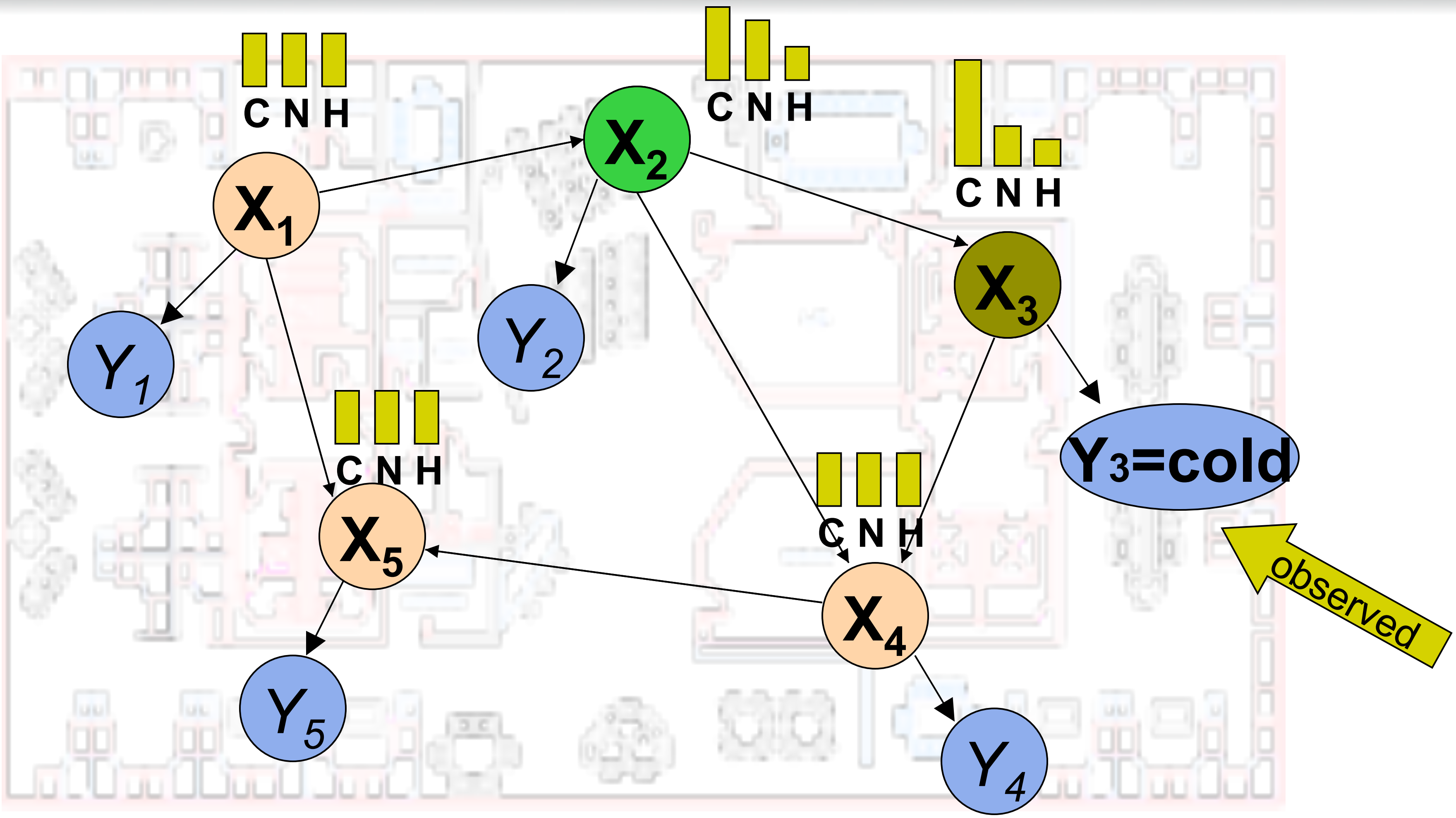
Making observations



Posterior entropy $H(\mathbf{X} \mid Y_3 = hot) \approx 2.7$

Reward: $H(\mathbf{X}) - H(\mathbf{X} \mid Y_3 = hot) \approx 1.5$

A different outcome…



Posterior entropy $H(\mathbf{X} \mid Y_3 = cold) \approx 3.2$

Reward: $H(\mathbf{X}) - H(\mathbf{X} \mid Y_3 = cold) \approx 1.0$

Information gain

- Entropy after observing Y = y:

$$H(X \mid Y = y) = -\sum_x P(X = x \mid Y = y) \log_2 P(X = x \mid Y = y)$$

- Don't know value of y before observing it!
- Conditional entropy:

$$H(X \mid Y) = \sum_y P(y) H(X \mid Y = y)$$

- Expected information gain (aka mutual information):

$$I(X; Y) = H(X) - H(X \mid Y)$$

Properties of entropy and infogain

Prod. rule: $P(X,Y) = P(X) \cdot P(Y|X)$

$H(X,Y) = H(X) + H(Y|X)$

$H(X_1 \ldots X_n) = H(X_1) + H(X_2|X_1) + H(X_3|X_{2,1}) + \ldots + H(X_n|X_1 \ldots X_{n-1})$

$H(X) \geq 0$

$\Rightarrow H(X) \geq H(X|Y)$ "information never hurts" (on average)

$I(X;Y) \geq 0$

$I(X;Y) = 0$ iff $X \perp Y$

$I(X;Y) = H(X) - \underbrace{H(X|Y)}_{H(X,Y)-H(Y)} = H(X) + H(Y) - \underbrace{H(X,Y)}_{H(Y,X)}$

$= I(Y;X)$

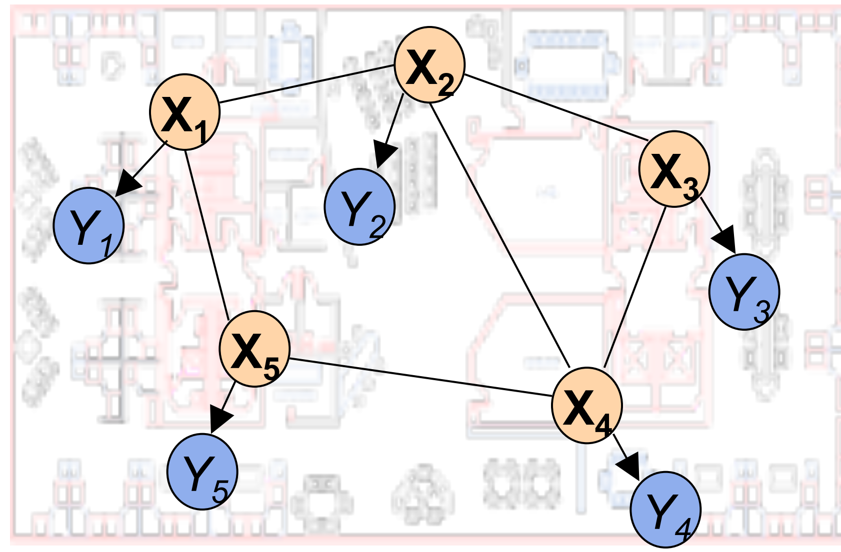
Maximizing information gain

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

$F(A) = I(X; Y_A)$

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

Typically NP-hard!



<u>Greedy algorithm:</u>

Start with A = {}

For i = 1 to k

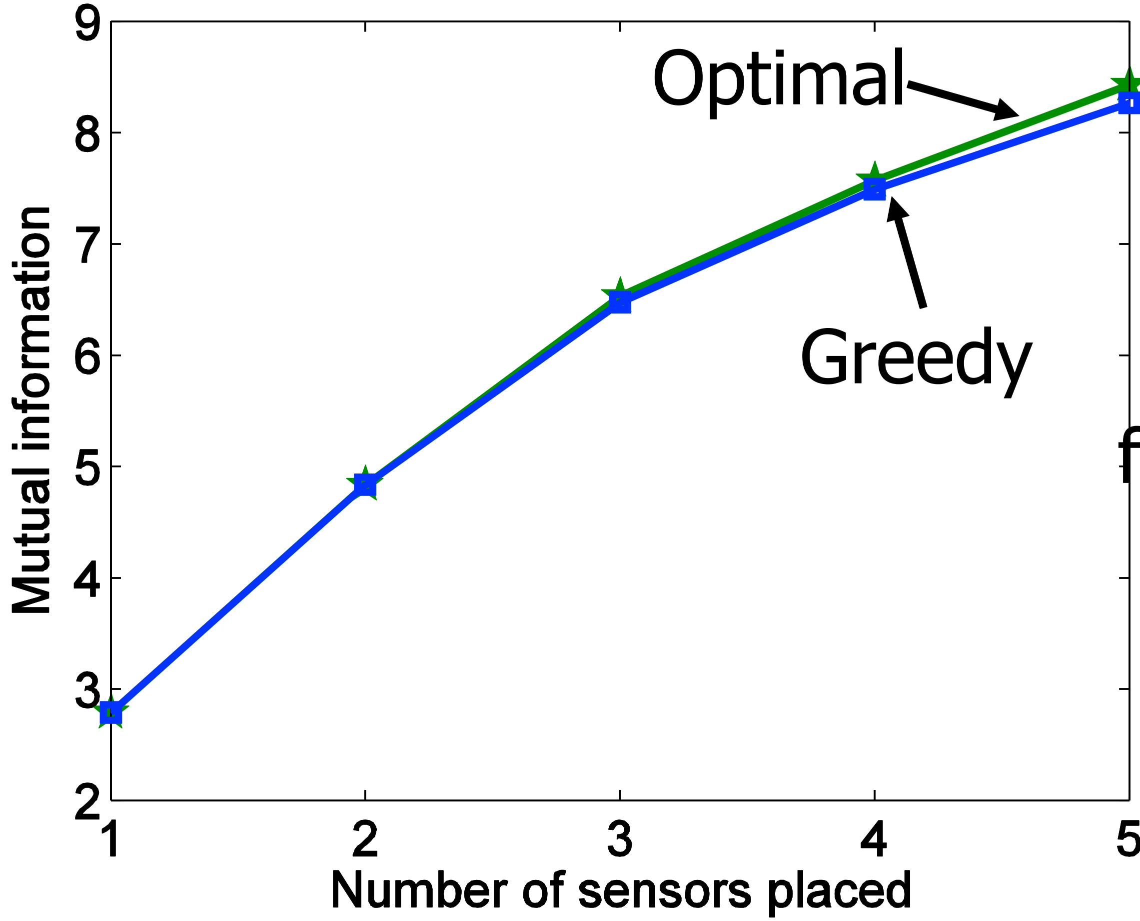
s* := argmax_s F(A U {s})

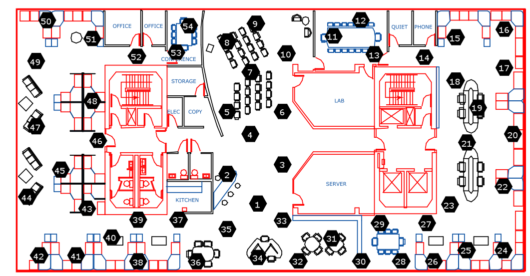
A := A U {s*}

How well can this simple heuristic do?

 37

Performance of greedy





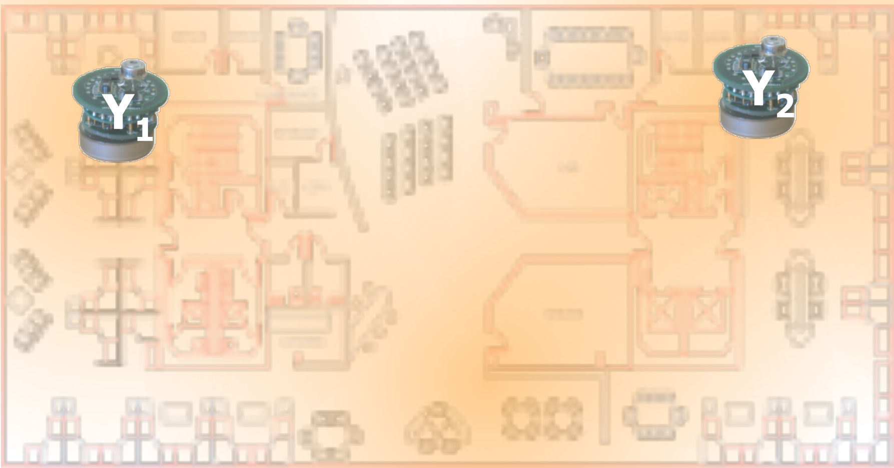
Temperature data from sensor network

- Greedy empirically close to optimal. Why?

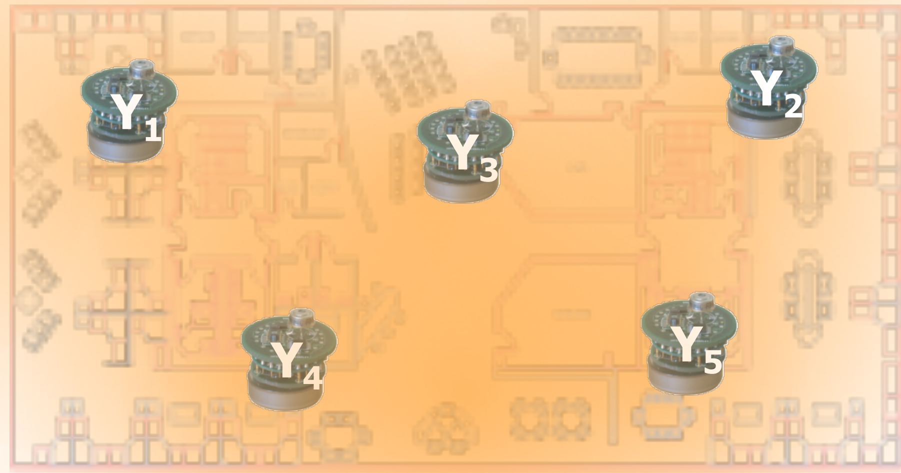
38

Key observation: Diminishing returns

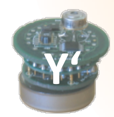
Placement A = $\{Y_1, Y_2\}$



Placement B = $\{Y_1, \ldots, Y_5\}$



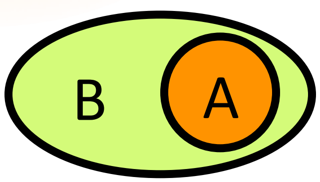
Adding Y' will help a lot!



Adding Y' doesn't help much

New sensor Y'

Submodularity:



\+ ● Y' Large improvement

\+ ● Y' Small improvement

For A µ B, F(A U {Y'}) – F(A) ≥ F(B U {Y'}) – F(B)

One reason submodularity is useful

Theorem [Nemhauser et al ‘78]

Greedy algorithm gives constant factor approximation

$$F(A_{greedy}) \geq (1\text{-}1/e)\ F(A_{opt})$$

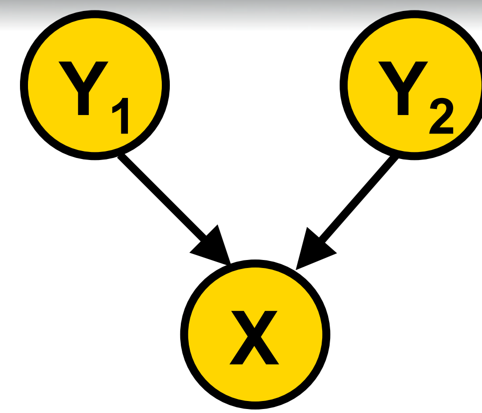
- Greedy algorithm gives near-optimal solution!
- Is information gain submodular?

 40

Non-submodularity of information gain

$Y_1, Y_2 \sim$ Bernoulli(0.5)

$X = Y_1$ **XOR** Y_2



Let $F(A) = I(Y_A; X) = H(X) – H(X|Y_A)$

$X \sim B(0.5)$ $\quad H(X) = 1$

$X \mid Y_1 = y \sim B(0.5)$ $\quad H(X|Y_1) = 1$

$X = Y_1$ XOR Y_2 $\quad H(X|Y_1, Y_2) = 0$

$H(X)$

$F(\{Y_1, Y_2\}) = H(X|Y_1, Y_2) = 1$

$F(\{\}) = H(X) - H(X) = 0$

$F(\{Y_1\}) = H(X) - H(X|Y_1) = 0$

— " — Y_2 — " — $Y_2 = 0$

↯

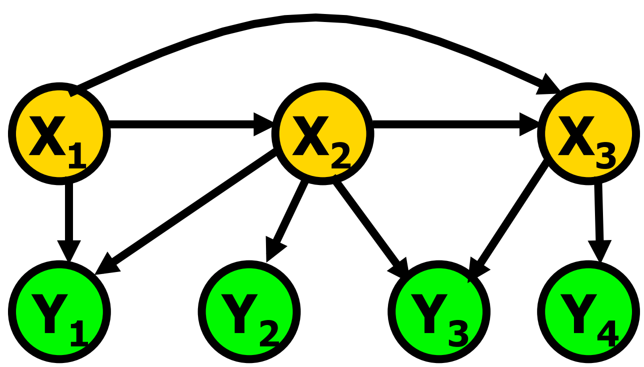
Example: Submodularity of info-gain

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

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

- However, NOT always submodular

Theorem
If Y_i are all conditionally independent given X, then F(A) is submodular!



Hence, greedy algorithm works!

In fact, NO algorithm can do better than (1-1/e) approximation!

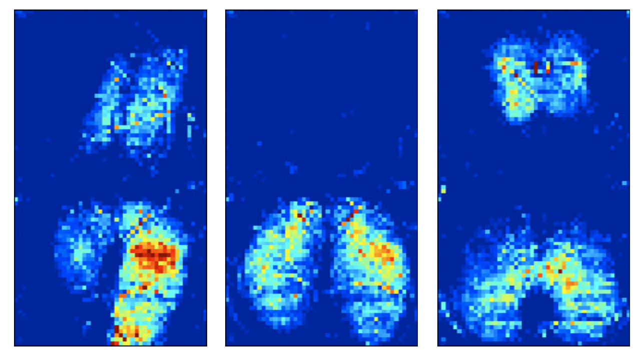
Case study: Building a Sensing Chair

- Activity recognition in assistive technologies
- Seating pressure as user interface

Equipped with 1 sensor per cm^2!

Costs $6,000!

Can we get similar accuracy with fewer, cheaper sensors?



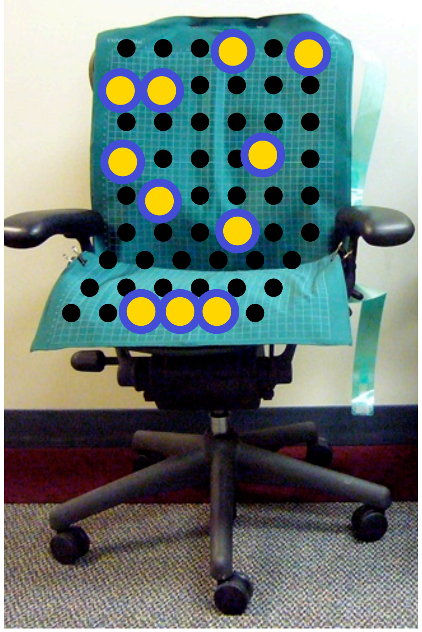
82% accuracy on 10 postures!

How to place sensors on a chair?

- Sensor readings at locations V as random variables
- Predict posture X using probabilistic model P(Y,V)
- Pick sensor locations A* µ V to minimize entropy:

$$\mathcal{A}^* = \underset{|\mathcal{A}| \leq k}{\operatorname{argmax}} \, I(X; \mathbf{Y}_{\mathcal{A}})$$

Possible locations **V**

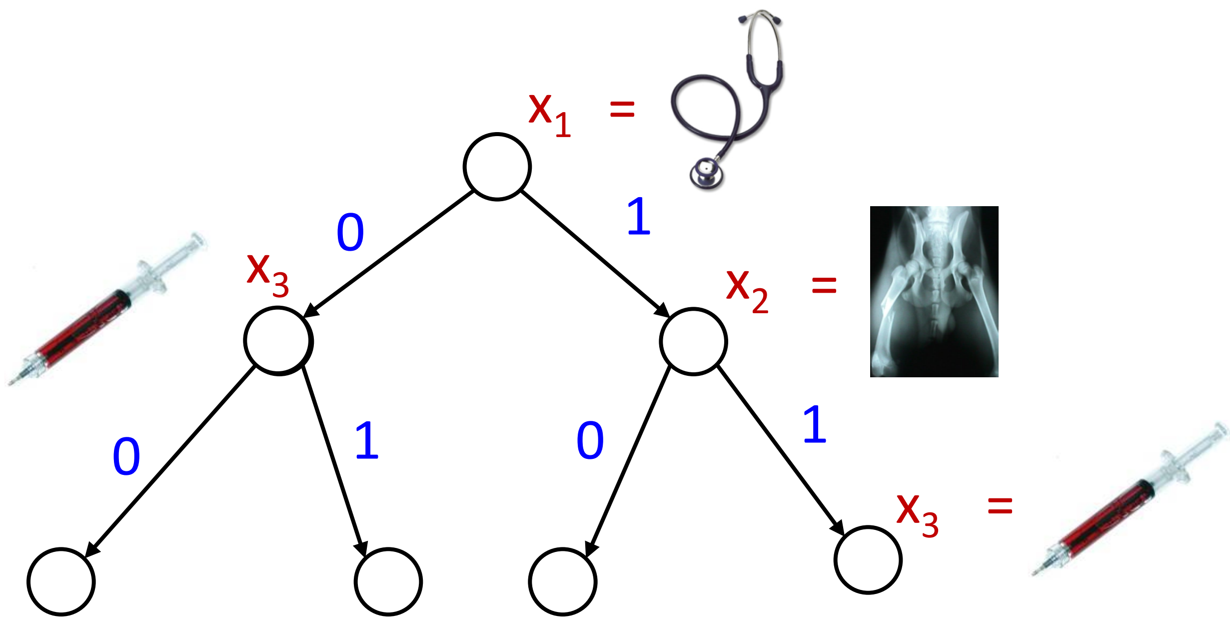


Placed sensors, did a user study:

	Accuracy	Cost
Before	82%	$6,000 ☹
After		

Adaptive Optimization

- So far: Search for a most informative *set* of variables (e.g., sensor placement).
- In many applications, want to adaptively choose observations:



Interested in a *policy* (decision tree), not a *set.*

Adaptive greedy algorithm

- Expected benefit of adding test s after we've seen $Y_A = y_A$.

$$\Delta(s \mid \mathbf{y}_A) = H(\mathbf{X} \mid \mathbf{y}_A) - \sum_{y_s} P(y_s \mid \mathbf{y}_A) H(\mathbf{X} \mid \mathbf{y}_A, y_s)$$

Adaptive Greedy algorithm:

Start with $A = \emptyset$

For i = 1:k

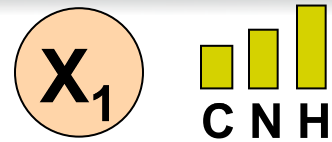
- Pick $s_k \in \arg\max_s \Delta(s \mid \mathbf{y}_A)$
- Observe $Y_{s_k} = y_{s_k}$
- Set $A \leftarrow A \cup \{s_k\}$

Gathering information for making decisions

- So far: Selecting variables which decrease the uncertainty the most
- Often, want to gather information to take the right action

Value of information

Should we raise a fire alert?



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

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

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

<u>Decision theoretic value of (perfect) information</u>

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

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

For a set A of variables, its expected *value of information* is

$$F(A) = \sum_{\mathbf{y_A}} \underbrace{P(\mathbf{y_A})}_{\substack{\text{Observations} \\ \text{made by sensors } \mathbf{A}}} \underbrace{MEU[\,\mathbf{X} \mid \mathbf{y_A}]}_{\substack{\text{Max. expected utility} \\ \text{when observing} \\ Y_A = y_A}}$$

Unfortunately, value of information is not submodular

Greedy algorithm can fail arbitrarily badly

Can do better with look-ahead

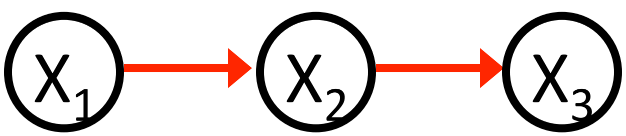
Maximizing value of information

[Krause, Guestrin '05]

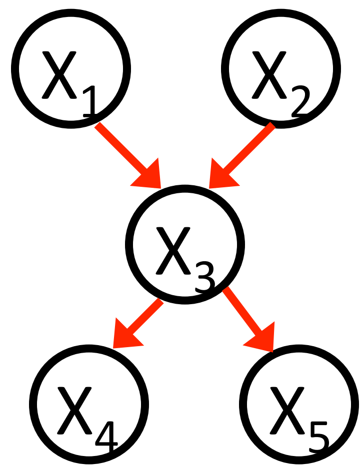
- Want to find a subset A^* of V, $|A^*| \leq k$ s.t.

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

Theorem: Complexity of optimizing value of information



For chains (HMMs, etc.)
Optimally solvable in polytime ☺



For trees:
NP^{PP} complete
☹