Active Learning and Optimized Information Gathering

Lecture 1 – Introduction

CS 101.2
Andreas Krause

Overview

- Research-oriented special topics course
- 3 main topics
  - Sequential decision making / bandit problems
  - Statistical active learning
  - Combinatorial approaches
- Both theory and applications
- Mix of lectures and student presentations
- Handouts etc. on course webpage
  - **http://www.cs.caltech.edu/courses/cs101.2/**
- Teaching assistant: Ryan Gomes (gomes@caltech.edu)

2

Background & Prerequisites

- Basic probability and statistics
- Algorithms

- Helpful but not required: Machine learning

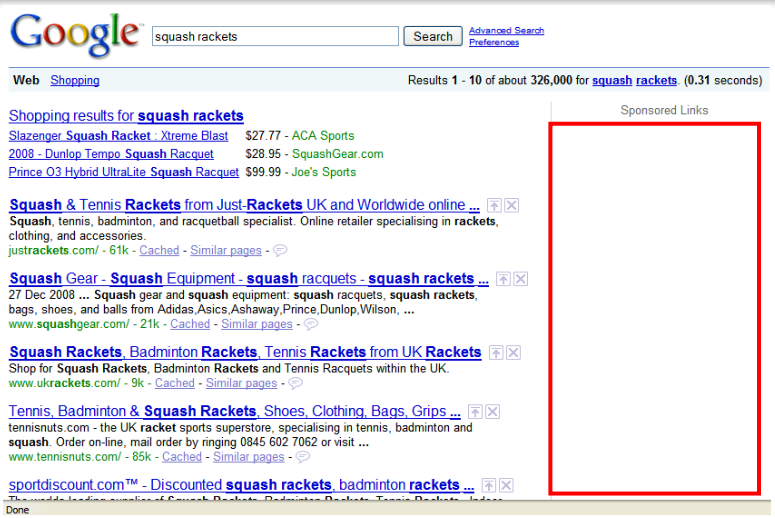
- Please fill out the questionnaire about background (not graded ☺)

3

How can we get **most useful** information at **minimum cost**?

4

Sponsored search



Which ads should be displayed to maximize revenue?

5

Sponsored search

- Earlier approaches: Pay by impression
  Go with highest bidder

  $\max_i q_i$

  ignores "effectiveness" of ads

- Key idea: Pay per click!
  Maximize revenue over all ads i

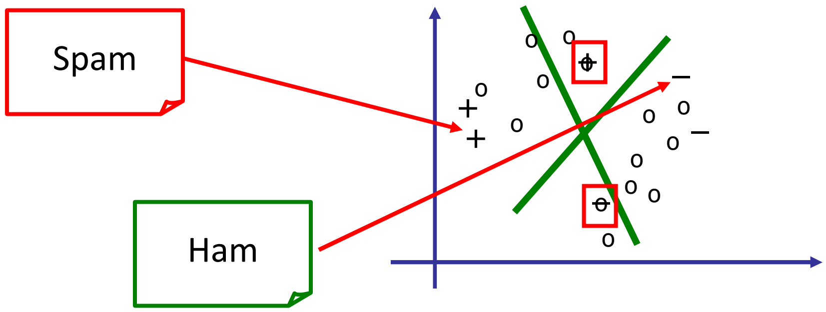
  $E[R_i] = P(C_i \mid \text{query})\, q_i$

Don't know! Need to gather information about effectiveness!

Bid for ad i (pay per click, known)

6

Spam or Ham?



- Labels are expensive (need to ask expert)
- **Which labels should we obtain to maximize classification accuracy?**

7

Clinical diagnosis?

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

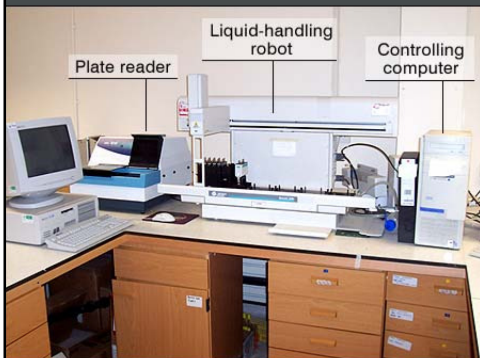
	healthy	*ill*
Treatment	-$$	$
No treatment	0	-$$$

- 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?**

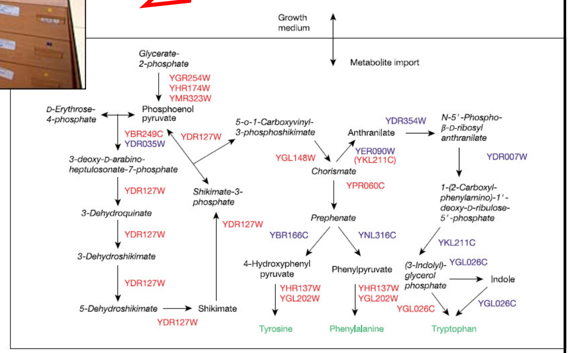
8

A robot scientist

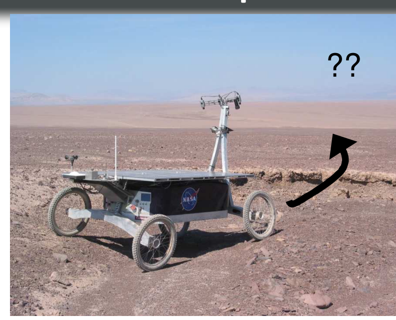


BBC

King et al, Nature '04

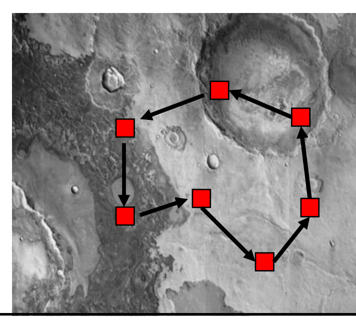


Autonomous robotic exploration

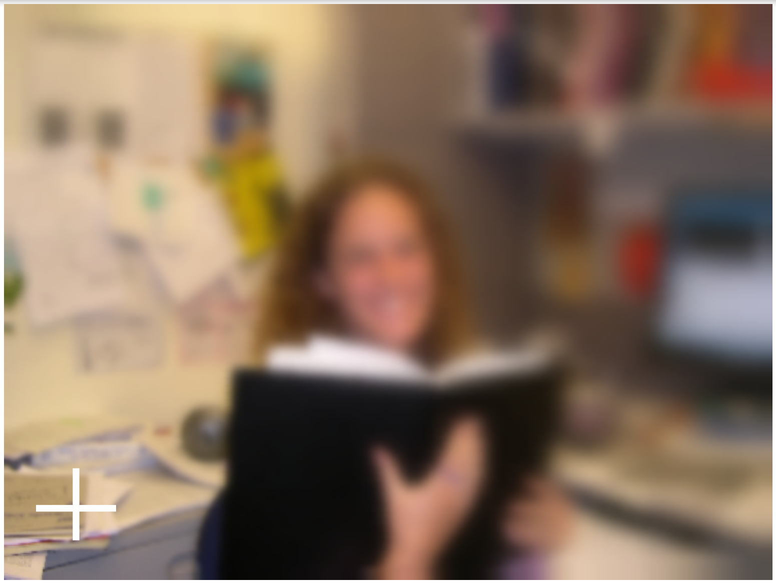


- Limited time for measurements
- Limited capacity for rock samples

- **Need optimized information gathering!**

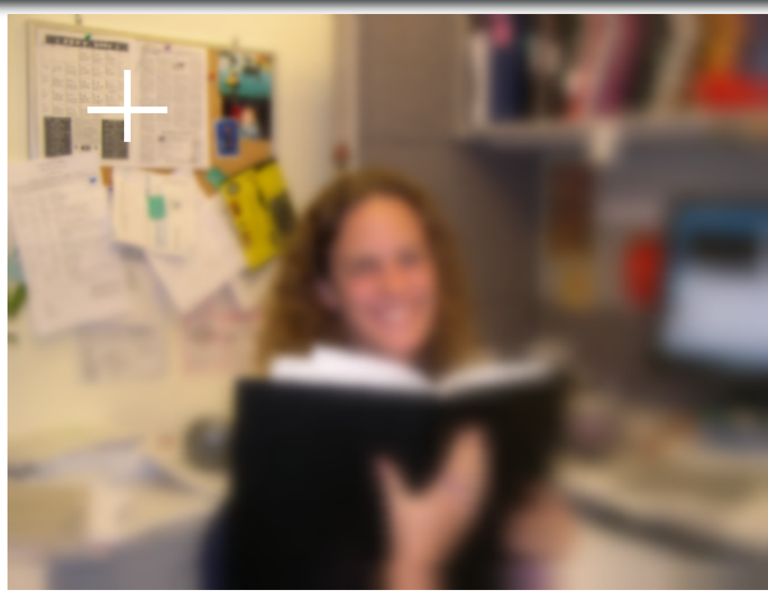


How do people gather information?
[Renninger et al, NIPS '04]



11

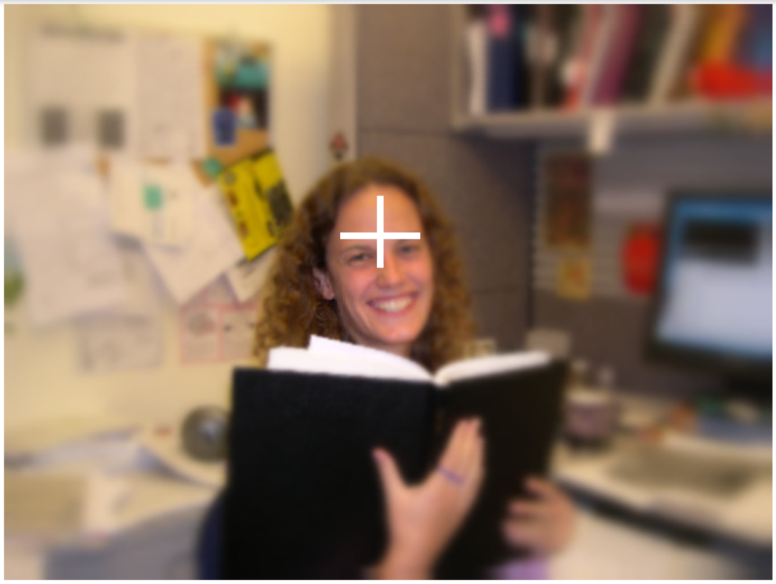
How do people gather information?
[Renninger et al, NIPS '04]



12

How do people gather information?

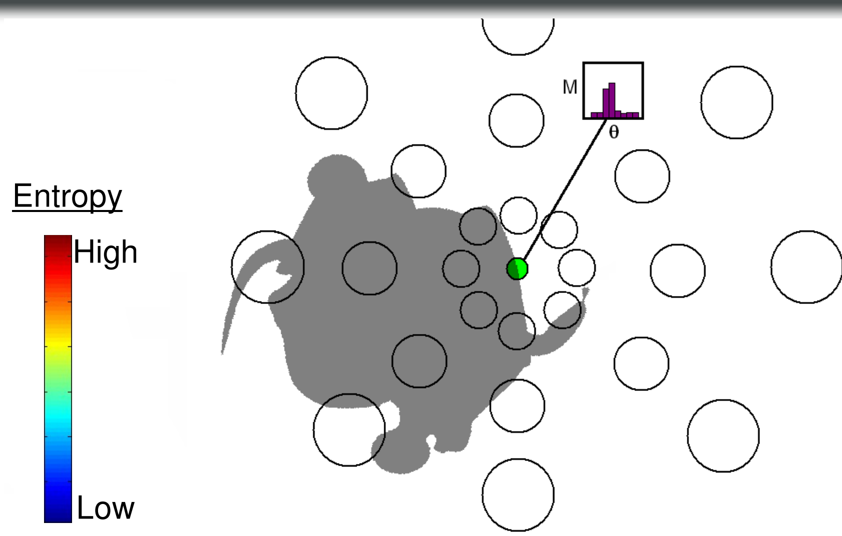
[Renninger et al, NIPS ’04]



13

How do people gather information?

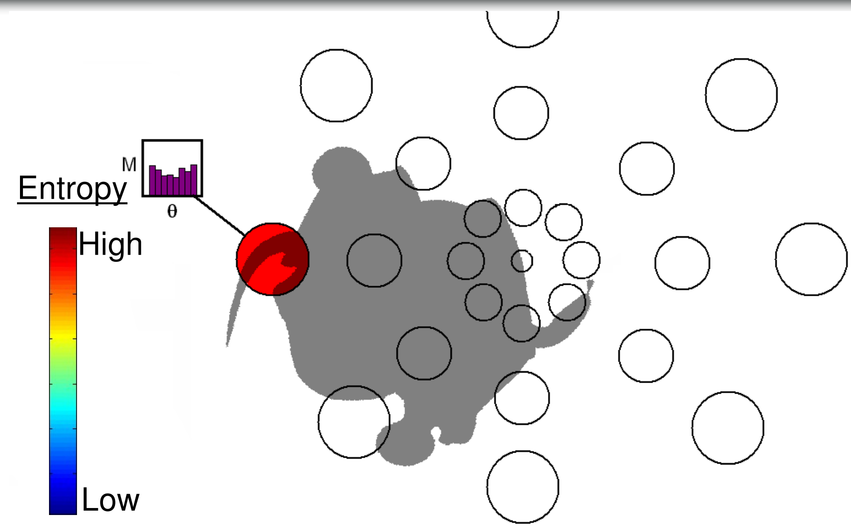
[Renninger et al, NIPS ’04]



14

How do people gather information?

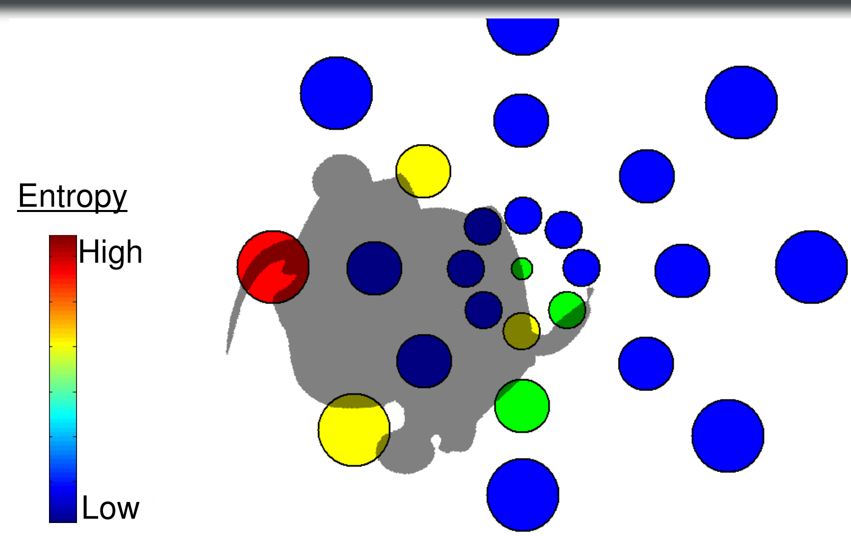
[Renninger et al, NIPS ’04]



15

How do people gather information?

[Renninger et al, NIPS ’04]



16

Key intellectual questions

- How can a machine choose experiments that allow it to maximize its performance in an unfamiliar environment?
- How can a machine tell "interesting and useful" data from noise?
- How can we develop tools that allow us to cope with the overload of information?
- How can we automate **Curiosity?**

17

Approaches we'll discuss

1. Online decision making
2. Statistical active learning
3. Combinatorial approaches

This lecture: Quick overview over all of them

18

What we won't cover

- Specific algorithms for particular domains
  - E.g., dialog management in Natural Language Processing
- Lots of heuristics without theoretical guarantees
  - We focus on approaches with provable performance
- Planning under partial observability (POMDPs)

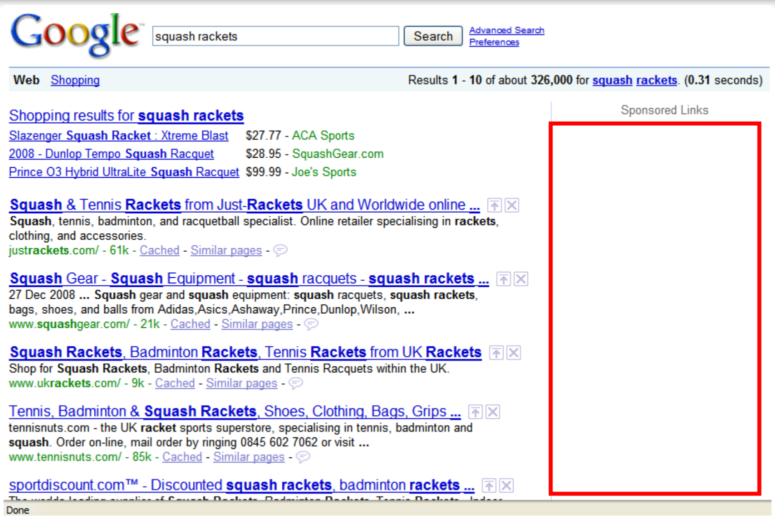
19

Approaches we'll discuss

1. **Online decision making**
2. Statistical active learning
3. Combinatorial approaches

20

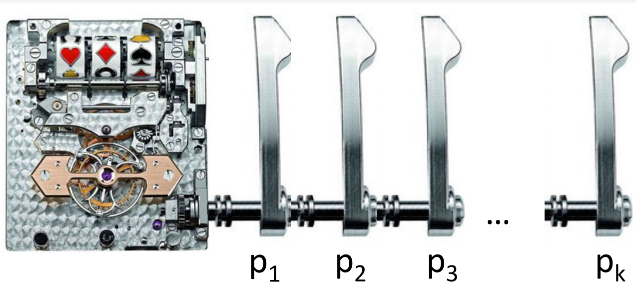
Sponsored search



Which ad should be displayed to maximize revenue?

21

k-armed bandits



- Each arm i
  - wins (reward = 1) with fixed (unknown) probability p_i
  - wins (reward = 0) with fixed (unknown) probability $1\text{-}p_i$
- All draws are independent given $p_1,\ldots,p_k$
- How should we pull arms to maximize total reward?

22

Online optimization with limited feedback

Choices	v_1				
a_1					
a_2	0				
…					
a_n					

Reward → Time

Total: $\sum_t v_t \rightarrow \max$

23

Performance metric: Regret

- Best arm: $p^* = \max_i p_i$
- Let $i_1,\ldots,i_T$ be the sequence of arms pulled
- Instantaneous regret at time t: $r_t = p^* - p_{i_t}$
- Total regret: $R = \sum_t r_t$

- Typical goal: Want pulling strategy that guarantees

$$R/T \rightarrow 0 \text{ as } T \rightarrow \infty$$

24

Arm pulling strategies

- Pick an an arm at random?

$p_1 \quad p_2 \quad p_3$

$0.1 \quad 0.5 \quad 0.9$

$\bar{r}_\pi = 0.4$

$v_1 \quad 1 \quad 0 \quad 0$

- Always pick the best arm?

Exploration—Exploitation Tradeoff

- **Explore** (random arm) with probability eps
- **Exploit** (best arm) with probability 1-eps

- Asymptotically optimal:

R = O(log T)

(More next lecture)

Bandits on the web

- Number of advertisements k to display is large
- Many ads are similar!
- Click-through rate depends on query
  - Similar queries → similar click-through rates!
  - Click probabilities depend on context
- Need to compile set of k ads (instead of only 1)

27

Bandit hordes

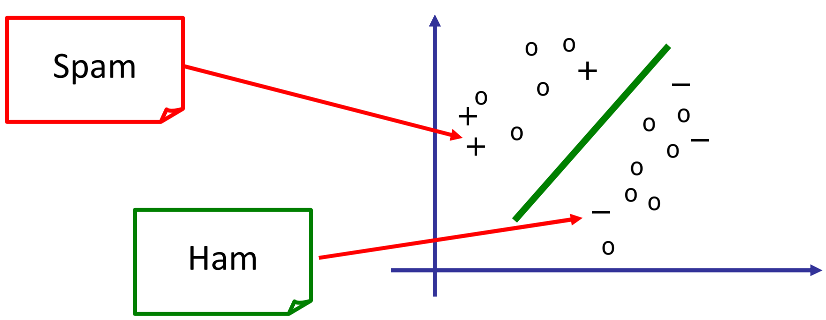
- k-armed bandits
- Continuum-armed bandits
- Bandits in metric spaces
- Restless bandits
- Mortal bandits
- Contextual bandits
- …

28

Approaches we'll discuss

1. Online decision making
2. **Statistical active learning**
3. Combinatorial approaches

Spam or Ham?

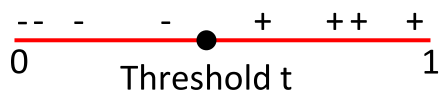


- Labels are expensive (need to ask expert)
- Which labels should we obtain to maximize classification accuracy?

Learning binary thresholds

- Input domain: D=[0,1]
- True concept c:

  c(x) = +1 if $x \geq t$
  c(x) = -1 if $x < t$

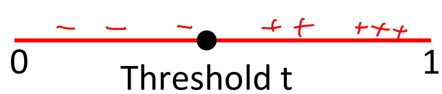


- Samples $x_1,\ldots,x_n \in D$ uniform at random

Passive learning

- Input domain: D=[0,1]
- True concept c:

  c(x) = +1 if $x \geq t$
  c(x) = -1 if $x < t$

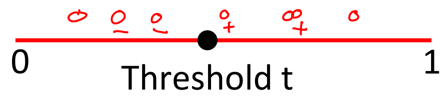


- Passive learning:
  Acquire all labels $y_i \in \{+,-\}$

Active learning

- Input domain: D=[0,1]
- True concept c:

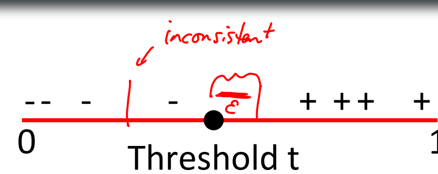
  c(x) = +1 if x$\geq$ t
  c(x) = -1 if x < t



- Passive learning:
  Acquire all labels $y_i \in$ {+,-}
- Active learning:
  Decide which labels to obtain

Classification error

- After obtaining n labels,
  D_n = {(x_1,y_1),…,(x_n,y_n)}
  learner outputs hypothesis consistent with labels D_n



- Classification error: R(h) = $E_{x\sim P}$[h(x) $\neq$ c(x)]

Statistical active learning protocol

Data source P (produces inputs x_i)

⇩

Active learner assembles data set

$D_n = \{(x_1,y_1),\ldots,(x_n,y_n)\}$

by selectively obtaining labels

⇩

Learner outputs hypothesis h

⇩

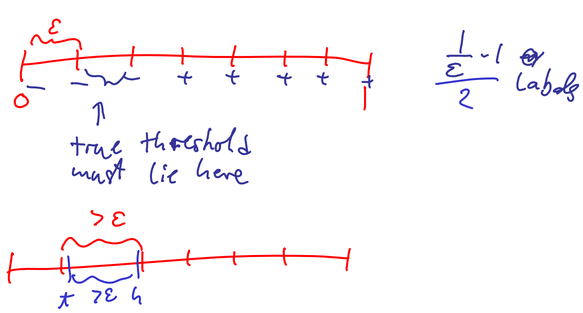
Classification error $R(h) = E_{x\sim P}[h(x) \neq c(x)]$

How many labels do we need to ensure that $R(h) \leq \varepsilon$?

35

Label complexity for passive learning

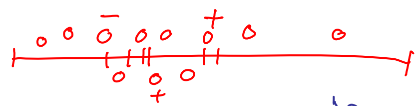
Allowable Classification error ε



Need at least $\Omega\left(\frac{1}{\varepsilon}\right)$ labels

36

Label complexity for active learning



"Binary search"
needs $O(\log \frac{1}{\varepsilon})$

37

Comparison

	Labels needed to learn with classification error ε
Passive learning	Ω(1/ε)
Active learning	O(log 1/ε)

Active learning can exponentially reduce the number of required labels!

38

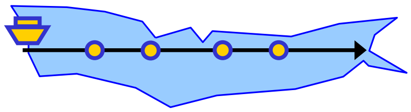
Approaches we'll discuss

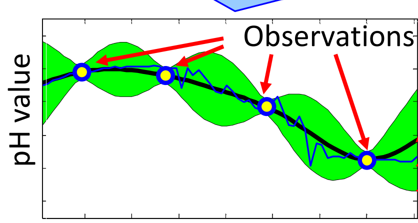
1. Online decision making
2. Statistical active learning
3. **Combinatorial approaches**

39

Automated environmental monitoring

- Monitor pH values using robotic sensor





Objective: $F(A) = \overbrace{H(V\setminus A)}^{\text{entropy of unobserved before observations}} - \underbrace{H(V\setminus A \mid A)}_{\text{entropy after making observations}}$

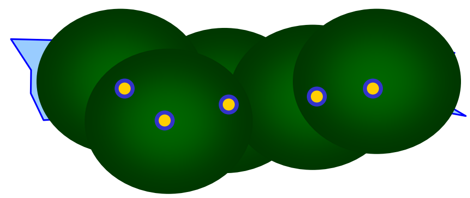
$\boxed{\text{Want } A^* = \text{argmax}_{|A| \leq k} F(A)}$

40

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

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

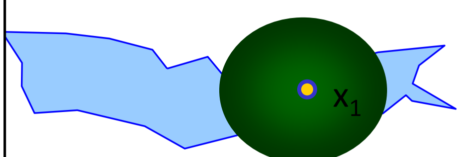
$A := A \cup \{s^*\}$

How well can this simple heuristic do?

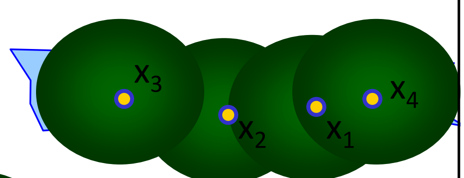
41

Key property: Diminishing returns

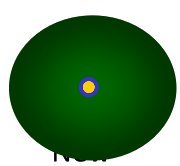
Selection $A = \{x_1\}$



Selection $B = \{x_1, x_2, x_3, x_4\}$



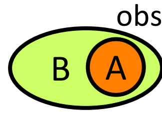
Adding x'
will help a lot!



New
observation x'

Adding x'
doesn't help much

Submodularity:



+ • s ⟸ Large improvement

+ • s ⟸ Small improvement

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

42

Why is submodularity useful?

Theorem [Nemhauser et al '78]

Greedy maximization algorithm returns A_{greedy}:

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

- Greedy algorithm gives near-optimal solution!
- Many other reasons why submodularity is useful
  - E.g.: Can solve more complex, combinatorial problems

43

What we've seen so far

- Optimizing information gathering is a challenging scientific question
- Taste for some of the tools that we have
  - Online optimization / bandit algorithms
  - Statistical active learning
  - Combinatorial approaches

44

Coursework

- Grading based on
  - Presentation (30%)
  - Course project (30%)
  - 3 homework assignments (one per topic) (30%)
  - Class participation (10%)
- Discussing assignments allowed, but everybody must turn in their own solutions
- Start early! ☺

45

Student presentations

- List of papers on course website
- By **tonight (January 6 11:59pm)**, pick an ordered list of 5 papers you'd be interested in presenting and email to krausea@caltech.edu
- Will get email with assigned paper and date by tomorrow

➔ Tentative schedule available Thursday

46

Presentation: Content

- Present **key idea** of the paper
- Do:
  - Introduce necessary terminology (reusing course notation whenever possible)
  - Visually illustrate main algorithm / idea if possible
  - Present high-level proof sketch of main result
  - Attempt to relate to what we've seen in the course so far
  - Clear presentation (not too crowded slides, etc.)
- Do NOT:
  - Attempt to explain every single technical lemma
  - Maximize the use of equations

47

Presentation: Format and Grading

- Presentation format up to you
  - PowerPoint, Keynote, LaTeX, Whiteboard, …
- After presentation, send slides to instructor (posted on course webpage)
- 35 Minutes + questions
- Grade based on
  - Presentation
  - Quality of slides / handouts
  - Answers to questions by students and instructor
- Evaluation sheet template on course webpage

48

Course project

- "Get your hands dirty" with the course material
- Implement the algorithm from the paper you presented (or some other paper) and apply it to some data set
- Ideas on the course website
- Application of techniques you learnt to your own research is encouraged

49

Project: Timeline and grading

- Small groups (2-3 students)
- January 20: Project proposals due (1-2 pages); feedback by instructor and TA
- January 27: Project start
- February 19: Project milestone
- March ~10: Project report due; poster session

- Grading based on quality of poster (20%), milestone report (20%) and final report (70%)

50

Tasks

- By tonight (January 6, 11:59pm): email to instructor
  - Ordered list of 5 papers
  - Questionnaire about background
- Start thinking about project teams and ideas (proposals due January 20)

51