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)
Background & Prerequisites

- Basic probability and statistics
- Algorithms
- Helpful but not required: Machine learning

Please fill out the questionnaire about background (not graded 😊)

How can we get most useful information at minimum cost?
Which ads should be displayed to maximize revenue?

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
\[ E[R_i] = P(C_i | \text{query}) q_i \]

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

Bid for ad i
(pay per click, known)
Spam or Ham?

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

Clinical diagnosis?

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

<table>
<thead>
<tr>
<th></th>
<th>healthy</th>
<th>ill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>-$$</td>
<td>$</td>
</tr>
<tr>
<td>No treatment</td>
<td>0</td>
<td>-$$</td>
</tr>
</tbody>
</table>

- 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?**
A robot scientist

Liquid-handling robot

Controlling computer

Plate reader

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]
How do people gather information?

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

Entropy
High  Low  

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

Approaches we’ll discuss

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

This lecture: Quick overview over all of them
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)

Approaches we’ll discuss

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

Which ad should be displayed to maximize revenue?

k-armed bandits

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

<table>
<thead>
<tr>
<th>Choices</th>
<th>$v_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$a_n$</td>
<td></td>
</tr>
</tbody>
</table>

Reward $\rightarrow$ Time

Total: $\sum_t v_t \rightarrow \text{max}$

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

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

- Pick an arm at random?

\[
\begin{bmatrix}
p_1 & p_2 & p_3 \\
0.1 & 0.5 & 0.9
\end{bmatrix}
\]

\[
\begin{bmatrix}
\nu_1 & 1 & 0
\end{bmatrix}
\]

- Always pick the best arm?

Exploration—Exploitation Tradeoff

- \textbf{Explore} (random arm) with probability \( \epsilon \)
- \textbf{Exploit} (best arm) with probability \( 1-\epsilon \)

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 $\Rightarrow$ similar click-through rates!
  - Click probabilities depend on context

- Need to compile set of $k$ ads (instead of only 1)

Bandit hordes

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

...
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, $D_n = \{(x_1,y_1),\ldots,(x_n,y_n)\}$
  learner outputs hypothesis consistent with labels $D_n$

- Classification error: $R(h) = \mathbb{E}_{x \sim \mathcal{P}}[h(x) \neq c(x)]$
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 $R(h) \leq \varepsilon$?

---

Label complexity for passive learning

All possible classification error $\varepsilon$

$\varepsilon$ threshold must lie here

Need at least $\Omega\left(\frac{1}{\varepsilon^2}\right)$ labels
Comparison

<table>
<thead>
<tr>
<th></th>
<th>Labels needed to learn with classification error $\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive learning</td>
<td>$\Omega(1/\varepsilon)$</td>
</tr>
<tr>
<td>Active learning</td>
<td>$O(\log 1/\varepsilon)$</td>
</tr>
</tbody>
</table>

Active learning can exponentially reduce the number of required labels!
Approaches we’ll discuss

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

Automated environmental monitoring

Monitor pH values using robotic sensor

- Position s along transect
- Observations A ⊆ V
- True (hidden) pH values
- Use probabilistic model (Gaussian processes) to estimate prediction error

Objective: $F(A) = H(V \setminus A) - H(V \setminus A | A)$

Want $A^* = \arg\max_{|A| \leq k} F(A)$
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

$$A^* = \arg \max_{|A| \leq k} F(A)$$

NP-hard!

Greedy algorithm:

Start with $A = \emptyset$
For $i = 1$ to $k$
    $s^* := \arg \max_s F(A \cup \{s\})$
    $A := A \cup \{s^*\}$

How well can this simple heuristic do?

Key property: Diminishing returns

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

Adding $x'$ will help a lot! Adding $x'$ doesn’t help much

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 $A_{\text{greedy}}$:
$$F(A_{\text{greedy}}) \geq (1 - 1/e) \max_{|A| \leq k} F(A)$$

~63%

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

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
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! 😊

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
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

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
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

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%)
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)