Lecture 16: Online Decision Making

Difficult is not just computational
but lack of information

Decision Making in the face of
Uncertainty

Approx + Lack of Info

Today
- Sleep + Tanam
- Competency

Next
- Online Learning
  - Blackwell & Hannan
  - regret
Online Agos

Optimal Search

Min \( \frac{\text{distance you travel}}{\text{optimal distance (in hindsight)}} \)

\[ \geq 2(1 + 3 + 5 + \ldots + d^*) \]

\[ \approx 2[d^*]^2 \]

Ratio \( \approx \frac{2[d^*]^2}{d^*} = 2d^* \to \infty \)
The diagram illustrates a problem involving distances and an optimization process. The key points are:

1. The distance is represented by $d^*$.
2. The problem involves finding the maximum distance traveled under certain constraints.
3. The solution involves an optimization process, denoted by OPT.
4. The objective is to minimize the distance $d^*$.
5. The expression for $d^*$ is given by $d^* = 2^c + 1$.
6. The inequality $\leq \frac{13}{9}$ suggests a constraint on the distance.
7. The term $4\left(1 + 2 + 4 + \ldots + 2^c\right) + 2^{m+2} + 2^c$ is derived and simplified to $4(2^m - 1) + 2^{m+1} + 2^c \leq 13.2^c$.
Def: \( (CR) \) competitive ratio of \( Ag \) of \( I \) = \( \frac{\text{E}[Ag(I)]}{\text{OPT}(I)} \) acting in partial info in hindsight

Worst case

Want \( Ag \) with best \( CR \)
**Number Guessing Game**

- **Algorithm**
  - **Binary Search**
    - Time complexity: $\log_2(N)$
    - $n \rightarrow 2 \log_2 n$
    - $\log_2 n + 2 \log_2 n$
    - $2 \log_2 n$
    - $O(\log_2 n)$

- **Analysis**
  - Lower Bound:
    - $\log_2 n + \log_2 (2n/3)$
    - $\frac{\log n}{\log_2 n}$
  - Upper Bound:
    - $2 \log_2 n$
    - $\log_2 n$
    - $O(\log_2 n)$
how many bits does $n$ take to represent?

1. Use dovetail search to learn $k$ in $2\lceil \log_2 k \rceil$ queries.
2. Use binary search to learn $n \in [2^{k/2}, 2^k]$ in $k = \log_2 n$ queries.

$\lceil \frac{\log_2 n}{2} \rceil + 2\lceil \log_2 \log_2 n \rceil$
Ski Rental:

- rent $1
- buy $B

Better Late Than Never:

Rent for B-1 days, then buy.

Theorem 1: \( CR(B|TN) = \max_T \text{ if } (T < B) \text{ then } T \text{ else } 2B-1 \) \( \min (T, B) \) = \( \frac{2B-1}{B} \) = \( 2 - \frac{1}{B} \).
Q: Can randomized alphas detect?  YES

Proposed algo:

- Flip a coin on day $B/2$.
- If heads, return.
- With heads probability $1/4$.
- If not, wait 2 and by a day $B/2$.

$\frac{e}{e-1} \approx 15$?
Caching (in System)

Eviction policy (when cache is full)

1. Furthest in future
   - optimal
   - requires future knowledge

2. LRU

3. FIFO (first in first out)

Cost: # of evictions
Every page in cache is marked/unmarked

All pages unmarked.

- When a page is requested, mark it (after bringing into cache).
- If you need to evict a page, evict a random unmarked page.
- If all pages are marked, unmark all.

Thm: Rand Mark algo is $O(\log K)$ competitive for caching problem.
Q: Show an algorithm that "end a random page" is only $k$-competitive.