Problem and Solution

CMPS 7010 Research Seminar

Research

The word *research* is derived from the Middle French "*recherche*", which means "to go about seeking", the term itself being derived from the Old French term "*recerchier*" a compound word from "re-" + "cerchier", or "sercher", meaning 'search'

- Basic research advances the fundamental knowledge about the world.
- Applied research is the practical application of science.

-- Wikipedia

The Problem

- A good problem is the heart of any high-quality research
- Focus on fundamentals
  - Refine your problem to remove trivialities
  - Ex: throughput vs. delay vs. complexity
- If you cannot solve a problem immediately
  - Save the partial result and revisit it when you have a new attack
  - It may help to think of two problems intermittently

Solution

- Sharp you skills
  - Taking a variety of courses: your last chance
  - Self-learning
- Develop your taste
  - Ask your advisor for examples of high-quality research
  - Read broadly and strategically
  - Search for elegance and insights

Solution – "PhD Research: Elements of Excellence"

- Do not be satisfied with a superficial result
  - Push the problem as far as you can
- There are no Gods in Academia
  - Learn to read papers written by good people
  - Read strategically & don’t become a clone of others’
  - Don’t be afraid of solving problems that other top researchers have looked at and failed or only partially succeeded.
- Never fall in love with the tool (or methodology)
  - Always remember: The problem is King

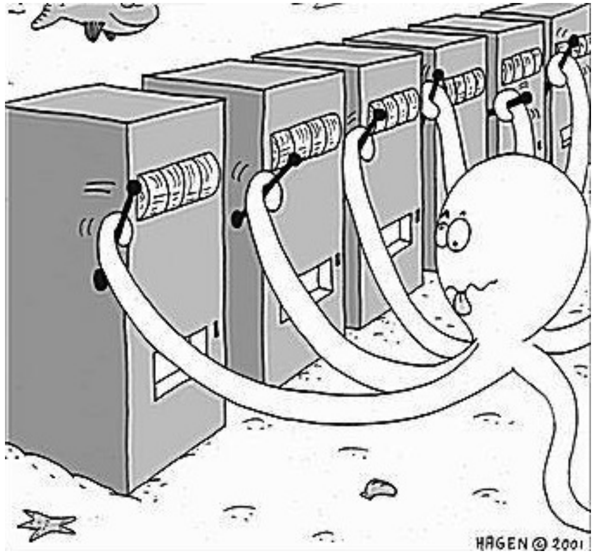
Strategies to attack a hard problem

- Exploit the unique structure of the problem
  - Ex: convexity, submodularity, ergodicity.
  - A solution that is independent of the problem structure is likely to be suboptimal
- Simplify the problem
  - Start with a special case or a toy example to get insights
  - The problem should still be non-trivial, and you have an attack

Strategies to attack a hard problem

- Find a "nearly optimal" solution
  - It is crucial to quantify "nearly"
  - Ex: approximation algorithms for NP-hard problem
- Settle for a less aggressive objective
  - Ex: regret in reinforcement learning, resource augmentation in online algorithms
- Alter the problem
  - You don’t have to work on the problem you are given - a key difference between math and engineering disciplines

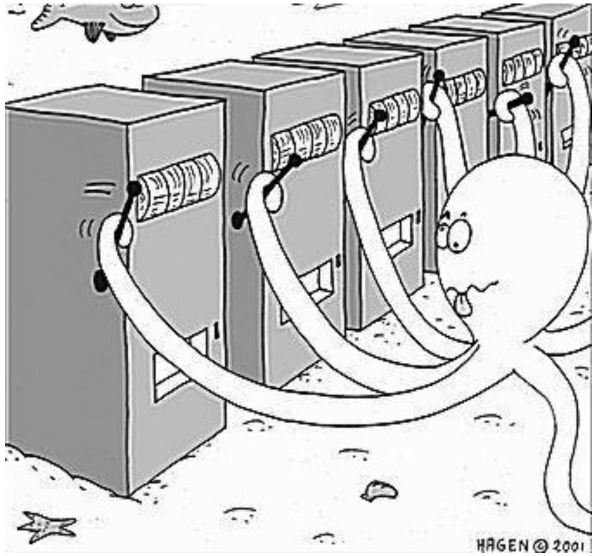
Case Study: Stochastic Multi-Armed Bandit



Case Study: Stochastic Multi-Armed Bandit

Given: K arms, T rounds. In each round $t \in T$:

1. Algorithm picks arm a_t.
2. Algorithm observes reward $r_t \in [0, 1]$ for the chosen arm



- The reward for arm a is $i.i.d.$ sampled from a distribution $\mathcal{D}_a$ that is initially unknown.
- Applications: news, ad selection, medical trial, etc.
- A fundamental tradeoff: exploration vs. exploitation

Chapter 1, “Introduction to Multi-Armed Bandits” by Aleksandrs Slivkins, 2019

Case Study: Stochastic Multi-Armed Bandit

Given: K arms, T rounds. In each round $t \in T$:

1. Algorithm picks arm a_t.
2. Algorithm observes reward $r_t \in [0, 1]$ for the chosen arm.

- Reward for arm a is $i.i.d.$ sampled from a distribution $\mathcal{D}_a$ that is initially unknown
- Let $\mu(a) = \mathbb{E}[\mathcal{D}_a]$, $\mu^* = \max_{a \in A} \mu(a)$

Regret: $R(T) = \mu^* \cdot T - \sum_{t=1}^{T} \mu(a_t)$

Objective: $\min \mathbb{E}[R(T)]$

Algorithm 1: Explore-First

1. Exploration phase: try each arm N times;
2. Select the arm a with the highest average reward (break ties arbitrarily);
3. Exploitation phase: play arm a in all remaining rounds.

Analysis for the 2-arm case:

- The regret in the exploration phase is trivially bounded by N
- The regret in the exploitation phase is determined by the probability that a is suboptimal. This can happen only if
  - (1) the mean rewards of the two arms are very close OR
  - (2) after 2N rounds, average reward is not close to mean reward for at least one arm

Algorithm 1: Explore-First

1. Exploration phase: try each arm N times;
2. Select the arm a with the highest average reward (break ties arbitrarily);
3. Exploitation phase: play arm a in all remaining rounds.

By taking $N = T^{2/3}$, $\mathbb{E}[R(T)] \leq T^{2/3} \times O(K \log T)^{1/3}$

- Poor performance in the exploration stage

Algorithm 2: Epsilon-Greedy

for each round $t = 1,2, \ldots$ do
  Toss a coin with success probability ϵ_t;
  if success then
    explore: choose an arm uniformly at random
  else
    exploit: choose the arm with the highest average reward so far
  end

By taking $\epsilon_t = t^{-1/3}(K\log t)^{1/3}$, $\mathbb{E}[R(t)] \leq t^{2/3} \times O(K \log t)^{1/3}$ for each round t

Adaptive Algorithms:

- A big flaw of Algorithms 1 and 2: exploration schedule does not depend on the observed rewards
- Adaptive algorithms
  - Successive Elimination
  - Optimism under uncertainty: UCB
  - Posterior sampling: Thompson sampling
  - ..

$$\mathbb{E}[R(T)] \leq O\left(\sqrt{KT\log T}\right)$$

- Can we do better?
  - Lower bound: fix T and K, there is a problem instance such that $\mathbb{E}[R(T)] \geq \Omega(\sqrt{KT})$