CSC 696H: Advanced Topics on AI **Fall 2024**

Lecture 8: LinUCB analysis; MDPs: Planning and Control

Lecturer: Chicheng Zhang *Scribe: (Tuan Nguyen & Sathvik Reddy Nookala)*

Introduction

This lecture focused on completing the analysis of LinUCB, a widely used algorithm in the field of contextual bandits, and introduced foundational concepts in reinforcement learning (RL). The lecture covered various exploration-exploitation strategies, the notion of episodic Markov Decision Processes (MDPs), and challenges like delayed consequences in decision-making scenarios.

Finish LinUCB

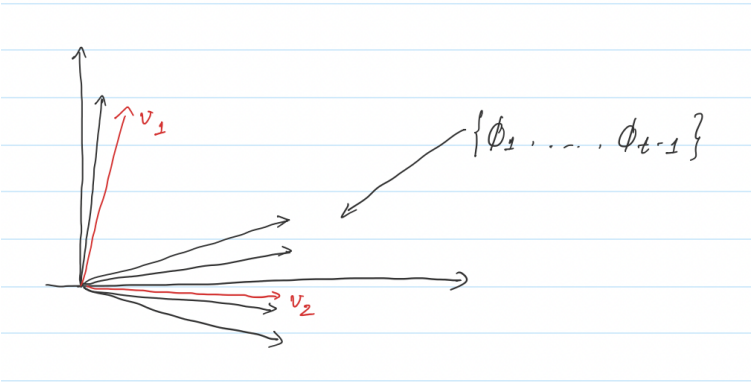
LinUCB's procedure

1. See context x_t
2. For each action a, compute $\text{UCB}_t(x_t, a) = \left\langle \hat{\theta}_t, \phi(x_t, a) \right\rangle + \beta_t ||\phi(x_t, a)||_{V_{t-1}^{-1}}$
   where the first term is the estimate reward of x_t and a
   and the second term is called exploration bonus of action a, denoted $b_t(a_t)$
3. take action $a_t = \text{argmax}_a \text{UCB}_t(x_t, a)$

Notation:

- $V_{t-1} = \sum_{s=1}^{t-1} \phi_s \phi_s^T + I$
- $\phi_s = \phi(x_s, a_s)$

Question: Imagine the historical data looks like the following figure with two new data vector v_1 and v_2, compare the novelty of $||v_1||^2_{V_{t-1}^{-1}}$ with $||v_2||^2_{V_{t-1}^{-1}}$



Answer: $||v_1||^2_{V_{t-1}^{-1}} \geq ||v_2||^2_{V_{t-1}^{-1}}$. Intuitively, v_2 is similar with 4 historical data while v_1 is similar with only 1 historical data. We did the calculation last lecture.

Analysis of LinUCB

- **Claim 1:** $\text{reg}_t \le 2b_t(a_t)$
- **Claim 2:** $\sum_{t=1}^{T} b_t(a_t) \le \tilde{O}(d\sqrt{T})$

Apply these two claims, we get $\text{Reg}(T) \le \tilde{O}(d\sqrt{T})$

Notation: $\tilde{O}$ hides any polypolylog(T)

Proof of Claim 2: We have

$$\begin{aligned}
\text{LHS} &= \sum_{t=1}^{T} \beta_t ||\phi(x_t, a_t)||_{V_{t-1}^{-1}} \\
&\le \tilde{O}(\sqrt{d}) \sum_{t=1}^{T} ||\phi(x_t, a_t)||_{V_{t-1}^{-1}} && \text{(use } \beta_t \le \tilde{O}(\sqrt{d})\text{)} \\
&\le \tilde{O}(\sqrt{d})\sqrt{T}\sqrt{\sum_{t=1}^{T} ||\phi_t||^2_{V_{t-1}^{-1}}} && \text{(use Cauchy–Schwarz inequality, } \phi_t = \phi(x_t, a_t)\text{)} \\
&\le \tilde{O}(\sqrt{d})\sqrt{T}\sqrt{\sum_{t=1}^{T} ||\phi_t||^2_{V_{t}^{-1}}} && \text{(use } V_{t-1}^{-1} \preceq V_t^{-1}\text{)}
\end{aligned}$$

Notation: $A \preceq B \Leftrightarrow B - A \succeq 0$, that means $B - A$ is a positive semidefinitie matrix

The Elliptic Potential Lemma (EPL)

Lemma 1. *Suppose* $u_1, \ldots, u_T \in \mathbb{R}^d$, $A_t = \sum_{s=1}^{t} u_s u_s^T + \mu I$, *then*

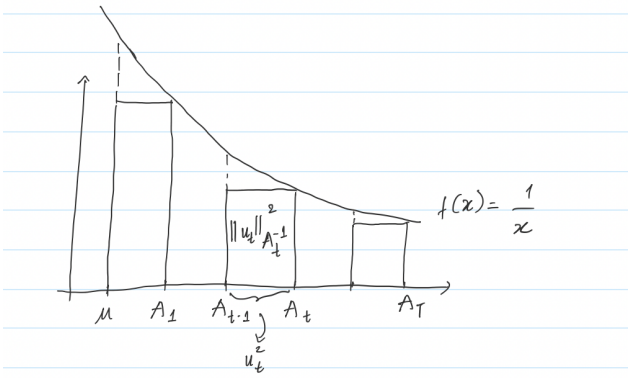
$$\sum_{t=1}^{T} ||u_t||^2_{A_t^{-1}} \le \ln\left(\frac{\det A_t}{\det A_0}\right) \le \tilde{O}(d \ln T)$$

where $A_0 = \mu I$

Intuition: In the case of $d = 1$, we have

- $A_t = \sum_{s=1}^{t} u_s^2 + \mu$
- $||u_t||^2_{A_t^{-1}} = u_t A_t^{-1} u_t = \frac{u_t^2}{\sum_{s=1}^{t} u_s^2 + \mu}$

Analyze $||u_t||^2_{A_t^{-1}} = \frac{u_t^2}{\sum_{s=1}^t u_s^2 + \mu}$



The area $||u_t||^2_{A_t^{-1}}$ is bounded by the area under the curve:

$$||u_t||^2_{A_t^{-1}} \leq \int_{A_{t-1}}^{A_t} \frac{1}{x} dx = \ln x \Big|_{A_{t-1}}^{A_t} = \ln \frac{A_t}{A_{t-1}}$$

Thus,

$$\sum_{t=1}^{T} ||u_t||^2_{A_t^{-1}} \leq \ln \frac{A_T}{A_0}$$

Similarly, for any d, we have

$$||u_t||^2_{A_t^{-1}} \leq \ln \left(\frac{\det A_t}{\det A_{t-1}} \right)$$

Applying the EPL with $\{u_t = \phi_t\}_{t=1}^T$ and $\mu = 1$, we have

$$\sum_{t=1}^{T} ||\phi_t||^2_{V_t^{-1}} \leq \tilde{O}(d \ln T)$$

Therefore,

$$\text{LHS} \leq \tilde{O}(\sqrt{d}) \sqrt{T} \sqrt{\tilde{O}(d \ln T)} = \tilde{O}(d\sqrt{T}) \quad \square$$

Final Remark

Triangle of Generalization, Exploration, and Delayed Consequences

Exploration: We studied Optimism principle. Other principles:

- Bayesian principle (Thompson Sampling)
- Estimation to Decision principle (E2D)

Delayed Consequences (Temperal Credit Assignment): (tabular) MDPs

Examples

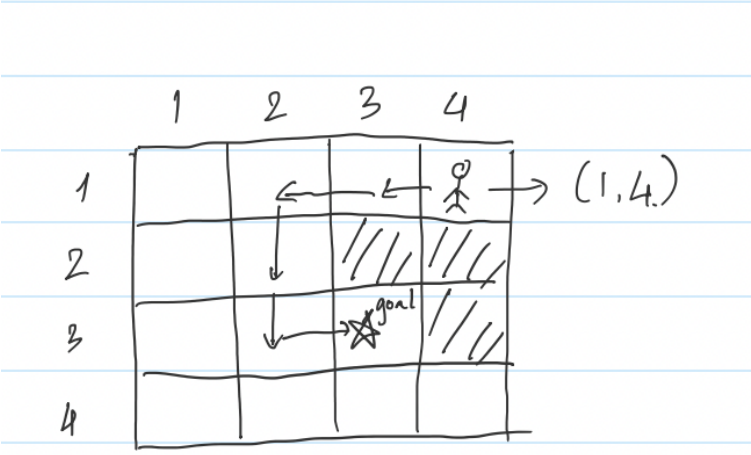
- chess
  - short term: we sacrifice a piece
  - long term: we are in favor position or win the game
- car maintenance
  - short term: we pay maintenance cost
  - long term: we do not have to fix major car issues

Finite Horizon, Episodic MDPs

Definitions

- $M = (\mathcal{S}, \mathcal{A}, H, (R_h)_{h=1}^H, (P_h)_{h=1}^H, \mu)$: the environment
- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- H: episode length
- $\forall h = 1, \ldots, H$: $R_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$: reward fn at step h
- $\forall h = 1, \ldots, H$: $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(S)$: transition probability at step h
  - $P_h(s'|s, a)$ where s' is the next state
- $\mu \in \Delta(S)$: initial state distribution

Example: Gridworld



The gridworld environment consists of

- $\mathcal{S} = \{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$
- $\mathcal{A} = \{L, R, U, D\}$
- an example for $(R_h)_{h=1}^H$
  - $R(s_{\text{goal}}, a) = 1$, $\forall a \in \mathcal{A}$
  - $R(s, a) = 0$, otherwise
- an example for $(P_h)_{h=1}^H$
  - determistic transition: when current position $s = (1, 4)$
  - $\forall h$, $P_h(s' = (1, 3)|s = (1, 4), a = \text{L}) = 1$
  - $\forall h$, $P_h(s' = (\ldots)|s = (1, 4), a = \text{L}) = 0$

Interaction in One Episode

In each episode of a finite horizon Markov Decision Process (MDP), the agent interacts with the environment over a fixed number of steps, denoted by H. The agent starts in an initial state, drawn according to an initial state distribution, and follows a sequence of actions to maximize the cumulative reward over the episode.

- **Initial State:** The agent observes the initial state s_1, which is sampled from the initial state distribution μ:

$$s_1 \sim \mu$$

- **Interaction Process:** For each time step $h = 1, 2, \ldots, H$, the agent performs the following:
  - Takes an action a_h.
  - Receives a reward $r_h = R_h(s_h, a_h)$, where R_h is the reward function at step h. The reward is deterministic and depends on the current state s_h and action a_h.

$$r_h = R_h(s_h, a_h)$$

  - Observes the next state s_{h+1}, which is drawn according to the transition probability P_h, conditioned on the current state and action:

$$s_{h+1} \sim P_h(s_{h+1}|s_h, a_h)$$

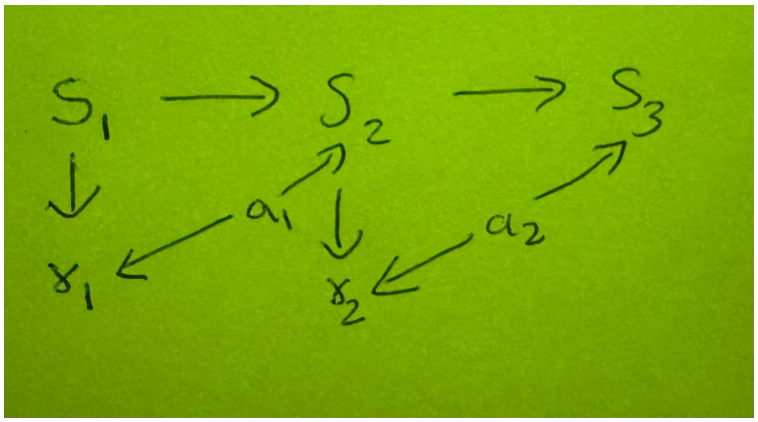
  The goal of the agent is to maximize the expected return, which is the cumulative reward over the episode:

$$\mathbb{E}\left[\sum_{h=1}^{H} s_h\right]$$

Note: The interaction follows a Markov structure, where the next state s_{h+1} depends only on the current state s_h and action a_h, making the process a Markov decision process (MDP).

Examples State Transitions Diagram

Here, we consider special cases of the episodic MDP when the episode length H is equal to 1.



- **Case 1:** $H = 1$
  - When the episode length H is 1, the agent only performs one action and observes one reward. This setup corresponds to **1 episode**, which is equivalent to **1 round of contextual bandit**. In a contextual bandit problem, the agent chooses an action in a single step based on the current context (state), with the objective of maximizing the immediate reward.
- **Case 2:** $H = 1$, $|S| = 1$
  - When both the episode length H is 1 and the state space contains only one state ($|S| = 1$), the problem reduces to **1 round of multi-armed bandit**. In this scenario, the agent has no dependence on any context or state, as there is only one possible state. The objective is to choose the best action (or arm) that maximizes the reward, which makes it equivalent to the classical multi-armed bandit problem.
- $H = 1 \Rightarrow 1$ episode $= 1$ round of contextual bandit.
- $H = 1, |S| = 1 \Rightarrow 1$ round of multi-armed bandit (MAB).

Example: Flying a Drone

In this example, we model the state and action spaces for flying a drone.

- **State:** $S = \begin{pmatrix} x \\ v \end{pmatrix}$
  - x: position
  - v: velocity
- **Action:** $a = u$ (force applied)
- **State Transition (Discrete Time):**

$$x_{t+1} = x_t + v_t$$
$$v_{t+1} = v_t + \frac{u_t}{m}$$

The transition in state can be written as:

$$\begin{pmatrix} x_{t+1} \\ v_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ v_t \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} u_t$$

This represents a linear dynamical system.

Cost Function (Negative Reward)
The cost function, or negative reward, is defined as:

$$C(s, a) = (x - x_0)^2 + \gamma u^2$$

Where:

- (x, v): s, a
- $(x - x_0)^2$: represents the position penalty.
- $\gamma * u^2$: represents the energy efficiency.

$$\text{Goal:} \quad \min \mathbb{E}\left[\sum_{t=1}^{T} C(s_t, a_t)\right]$$

Where:

- $C(s_t, a_t)$ is the cost function at time step t, which is generally quadratic in both the state s_t and the action a_t.
- T is the time horizon over which we are minimizing the cumulative cost.
- The system is assumed to follow the Linear Quadratic Regulator (LQR) framework, where both the system dynamics and the cost function are linear and quadratic, respectively.

This is a standard optimization problem for linear dynamical systems, where the controller aims to regulate the system efficiently while minimizing a quadratic cost.

Variant(Infinite Horizon Discounted Setting)

In the infinite horizon setting, the agent interacts with the environment indefinitely. To ensure convergence of returns, we introduce a discount factor $\gamma \in [0, 1)$, which makes rewards obtained further in the future less valuable.

- **Horizon:** $H = \infty$. The time horizon is infinite, meaning the agent will interact with the environment over an infinite number of steps.
- **Reward Function:** $R_n \equiv R$ for all n. The reward function R is the same for all steps n.
- **Transition Function:** $P_n \equiv P$ for all n. The transition probability function P is identical at every step.
- **Discount Factor:** $\gamma < 1$. The discount factor γ reduces the value of future rewards. It ensures that the agent values immediate rewards more than future rewards.
- **Goal:** Maximize the expected discounted return:

$$\mathbb{E}\left[\sum_{h=1}^{\infty} r_h \cdot \gamma^{h-1}\right]$$

Here, r_h represents the reward at step h, and γ^{h-1} is the discount applied to the reward received at step h.

- **Effective Horizon:**

$$\text{Effective Horizon} = \frac{1}{1-\gamma}$$

  The effective horizon quantifies the time scale over which rewards significantly contribute to the return. The closer γ is to 1, the longer the effective horizon, meaning future rewards are almost as valuable as immediate rewards.

- The discounted return can be written as:

$$r_1 + \gamma \cdot r_2 + \gamma^2 \cdot r_3 + \gamma^3 \cdot r_4 + \cdots$$

Delayed Consequences Problem in MDP

In this section, we discuss the **Delayed Consequences** problem in the context of a Markov Decision Process (MDP). Specifically, we analyze the so-called **"Combo lock"** MDP, which demonstrates the phenomenon where an agent's actions have delayed effects on rewards.

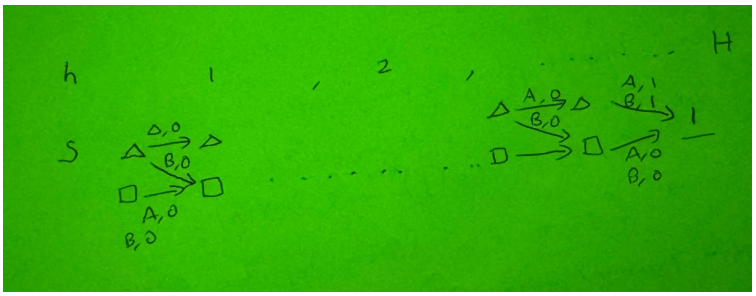
MDP Setup:

The MDP has the following components:

- **State space:** $S = \{\triangle, \square\}$. The agent can be in one of two states, represented by $\triangle$ (triangle) or $\square$ (square).
- **Action space:** $A = \{A, B\}$. At each step, the agent can choose between two actions, labeled A and B.
- **Initial state distribution:** $\mu = (1, 0)$. The agent starts in state $\triangle$ with probability 1.
- **Transition and reward functions:** The transition and reward functions $(R_h, P_h)_{h=1}^{H}$ are deterministic. This means that given a state and action at any step h, the resulting state and reward are fixed and do not involve any randomness.

State-Action Transition Diagram:

The following diagram illustrates how states and actions evolve over time:



Key Observations:

1. To get a non-zero reward, the agent needs to follow a specific action sequence. In this case, the agent must take the action sequence $A_1, A_1, \ldots, A_1$ in successive steps. Any deviation from this sequence results in zero reward.

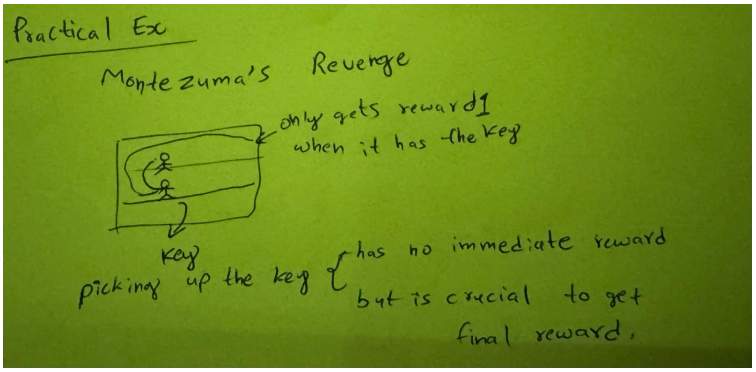
2. Consider the action sequence $BA \ldots A$. The agent will receive a reward of 0 (zero).

   - Action B has what we call **delayed consequences**. Although taking action B does not lead to any immediate difference in reward, it puts the agent in a "bad" state. The bad state leads to lower future rewards, even if the agent takes the correct actions afterward.

This example demonstrates how certain actions, although seemingly inconsequential in the short term, can have a profound impact on future rewards in an MDP.

Practical Example: Montezuma's Revenge

In this example, we discuss the delayed consequences in the Montezuma's Revenge problem. The agent in the game only gets the reward when it possesses the key. However, picking up the key does not have an immediate reward, but it is a crucial action to eventually achieve the final reward.



Optimal Behavior and Policies

Definition of History-Dependent Policy

A **history-dependent policy** π is a collection of functions π_h for $h = 1, 2, \ldots, H$. The policy at each step depends on the full history of states, actions, and rewards up to step h, as follows:

$$\pi_h \left(a_h \mid (s_1, a_1, r_1, \ldots, s_{h-1}, a_{h-1}, r_{h-1}, s_h)\right) = \Delta(A)$$

In this formulation: - π_h selects an action a_h based on the history of past states $s_1, \ldots, s_{h-1}$, actions $a_1, \ldots, a_{h-1}$, and rewards $r_1, \ldots, r_{h-1}$, as well as the current state s_h. - The policy outputs a distribution over actions A.

Markovian (Reactive) Policy

A **Markovian (reactive) policy** π^M is a simplified form of a policy, where the action at each step h only depends on the current state s_h, rather than the full history:

$$\pi_h = \pi_h(a_h \mid s_h)$$

This is a memoryless (Markovian) policy, where the decision at step h only considers the present state s_h, not the sequence of previous states and actions. If $\pi_h : S \to \Delta(A)$, the policy is stochastic, and if $\pi_h : S \to A$, the policy is deterministic.

Expected Return of a Policy

Given an MDP M and a policy π, the two together determine a distribution over possible trajectories $\mathcal{T}$ (sequences of states, actions, and rewards).

The expected return $J(\pi)$ of a policy π is the expected cumulative reward over an episode:

$$J(\pi) = \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]$$

This is calculated as:

$$J(\pi) = \sum_{\mathcal{T}} P(\mathcal{T} \mid \pi)\left(\sum_{h=1}^{H} r_h\right)$$

Where: - $\mathcal{T}$ is a trajectory of the MDP. - $P(\mathcal{T} \mid \pi)$ is the probability of a trajectory occurring under policy π. - r_h is the reward at step h.

Planning and Optimal Control

Given an MDP M, the goal is to find an optimal policy $\pi^* \in \Pi$ that maximizes the expected return:

$$\pi^* = \arg\max_{\pi \in \Pi} J(\pi)$$

This involves searching over the set of all possible policies Π to identify the one that maximizes the expected return.

—

This concludes the discussion on the delayed consequences problem in the Montezuma's Revenge game and optimal behavior in MDPs.