CSC 588: Machine learning theory **Fall 2024**

Lecture 6: Exploration and Optimization in Bandit Algorithms

Lecturer: Chicheng Zhang *Scribe: Razvan-Gabriel Dumitru*

Introduction

In the realm of reinforcement learning, bandit algorithms represent a critical area of study that focuses on balancing the trade-off between exploration and exploitation. These algorithms are fundamental to understanding how decisions can be optimized in uncertain environments. This presentation delves into various types of bandit algorithms, including multi-armed bandits (MAB) and stochastic linear bandits (SLB), outlining their core principles, mathematical formulations, and practical applications. We explore the optimization strategies inherent in the Upper Confidence Bound (UCB) techniques and linear contextual models, examining how these approaches can be effectively applied to real-world problems through rigorous data analysis and parameter estimation.

1 Multi-Armed bandits

Recap

The Multi-Armed Bandit (MAB) problem is a decision-making framework in which an agent selects from a set of actions (or ”arms”) to maximize cumulative reward over time. The agent must balance exploration, selecting less-known actions to gather information, and exploitation, choosing actions believed to yield the highest rewards. The goal is to minimize the regret, defined as the difference between the cumulative reward of the optimal strategy and the chosen actions over a time horizon T.

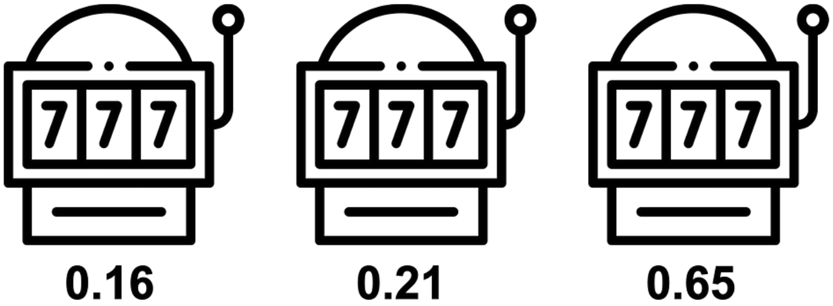

Figure 1: Example of MAB for slots.

Figure 1 shows a set-up with three slots machine all with a given probability, for each one of those probabilities we add noise to represent a real case scenario.

Actions: $\mathcal{A} = \{1, \ldots, A\}$

The **Reward** when the agent takes action a (pulls arm a) can be written as: $r = f(a) + \epsilon$, where:

- $f(a)$: expected reward of action a
- ϵ: noise, which we assume to be independent, zero-mean, 1-subgaussian

The **Regret** can be written as: $T \cdot f^*(a^*) - \mathbb{E}\left[\sum_{t=1}^{T} f^*(a_t)\right] = \mathbb{E}[\sum_{t=1}^{T}(f^*(a^*) - f^*(a_t))]$, where

- $f(a^*)$: optimal expected reward
- $f^*(a^*) - f^*(a_t)$: instantaneous regret

As a reminder of possible MAB algorithms and their corresponding Regrets we have the following list:

- Explore then Commit: $O(A^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$
- ϵ-greedy: $O(A^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$
- Upper confidence bound (UCB): $O(\sqrt{AT})$

The UCB Algorithm

The UCB (Upper Confidence Bound) algorithm is designed for decision-making in uncertain environments by balancing exploration and exploitation. At each time step t, the algorithm selects an action a_t that maximizes the upper confidence bound.

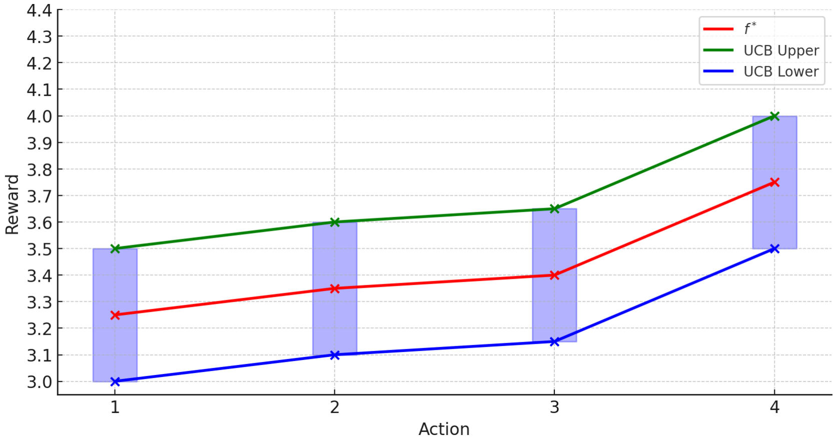

Figure 2: This plot illustrates the relationship between the true reward function f^* (red line) and the estimated Upper and Lower Confidence Bounds (green and blue lines, respectively) for different actions (1 through 4). The shaded areas represent the confidence intervals around each action's reward estimation. The true function f^* is deliberately positioned in the middle of these intervals to show how the confidence bounds encapsulate the range of plausible reward values given the uncertainty in estimation.

At time step t:

$$a_t = \arg\max_{a \in A} UCB_t(a)$$

where $\mathcal{A} = \{1, \ldots, A\}$.

The $UCB_t(a)$ is defined as:

$$UCB_t(a) = \hat{f}_t(a) + b_t(a)$$

where:

- $\hat{f}_t(a)$ is the mean reward of action a (exploitation).
- $B_t(a)$ is the ”bonus”, used to balance exploitation and exploration, given by:

$$b_t(a) = \sqrt{\frac{l}{N_t(a) + 1}}; l = 8\ln(2AT)$$

where $N_t(a)$ is the number of times action a has been selected up to time t.

Analysis of UCB

Regret Analysis

The expected regret is given by:

$$\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}(\text{reg}_t)\cdot I(E)\right]}_{(1)}+\underbrace{\mathbb{E}\left[\sum_{t=1}^{T}(\text{reg}_t)\cdot I(E^c)\right]}_{\leq 2}$$

We aim to show that:

$$\mathbb{E}\left[\sum_{t=1}^{T}(\text{reg}_t)\cdot I(E)\right]\leq 2.$$

Event Definition

Define event E such that for all t and a,

$$|\hat{f}_t(a)-f^*(a)|\leq b_t(a)$$

In our last lecture, we presented a lemma (without proof) that shows that $\mathbb{P}(E)\geq 1-\frac{2}{T}$ and $\mathbb{P}(E^c)\leq\frac{2}{T}$.

Bounding (1)

To bound (1), we aim to upper bound $\sum_{t=1}^{T}\text{reg}_t$ when E happens.

Claim 1: For any t, the regret is bounded by:

$$\text{reg}_t\leq 2b_t(a_t)$$

Proof:

$$\begin{aligned}\text{reg}_t&=f^*(a^*)-f^*(a_t)\\&\leq UCB_t(a^*)-f^*(a_t)\quad\text{(Validity of UCBs at }E)\\&\leq UCB_t(a_t)-f^*(a_t)\quad\text{(As }a_t\text{ maximizes UCB)}\\&\leq 2b_t(a_t)\quad\text{(Since }f^*(a_t)\in[LCB_t(a_t),UCB_t(a_t)]\text{,where }LCB\text{ is the lower-bound)}\end{aligned}$$

Claim 2:

$$\sum_{t=1}^{T}b_t(a_t)\leq 4\sqrt{l}\cdot\sqrt{AT}$$

(Proving this will conclude the UCB analysis)

We can see what happens in practice to b_t in the following table:

t	1	2	3	4	5	6
a_t	1	1	2	1	2	1
$b_t(a_t)$	$\underbrace{\sqrt{\frac{l}{1}}}_{N_0(1)+1}$	$\underbrace{\sqrt{\frac{l}{2}}}_{N_1(1)+1}$	$\underbrace{\sqrt{\frac{l}{1}}}_{N_2(2)+1}$	$\underbrace{\sqrt{\frac{l}{3}}}_{N_3(1)+1}$	$\underbrace{\sqrt{\frac{l}{2}}}_{N_4(2)+1}$	$\underbrace{\sqrt{\frac{l}{4}}}_{N_5(1)+1}$

Proof

$$\begin{aligned}
\sum_{t=1}^{T} b_t(a_t) &= \sum_{a=1}^{A} \sum_{\substack{t=1\\ a_t=a}}^{T} b_t(a) \text{ for } N_T(a) \text{ turns} \\
&= \sum_{a=1}^{A} \sum_{i=1}^{N_T(a)} \sqrt{\frac{l}{i}} \quad \left(\text{Fact: } \sum_{n=1}^{N} \frac{1}{\sqrt{n}} \leq 2\sqrt{N}\right) \\
&= 4\sqrt{l} \cdot \sum_{a=1}^{A} \sqrt{N_T(A)} = A\left(\frac{1}{A}\sum_{a=1}^{A} \sqrt{N_T(a)}\right)
\end{aligned}$$

Application of Jensen's Inequality

Since $\sqrt{x}$ is a concave function, thus $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$, by Jensen's inequality, we have:

$$A\left(\frac{1}{A}\sum_{a=1}^{A} \sqrt{N_T(A)}\right) \leq A \cdot \sqrt{\frac{1}{A}\sum_{a=1}^{A} N_T(a)} = A \cdot \sqrt{\frac{T}{A}} = \sqrt{AT}$$

Since we know that:

$$\sqrt{a} + \sqrt{b} \leq 2\sqrt{\frac{a+b}{2}} \quad (\text{for } a, b \geq 0)$$

Recipe for Designing MAB Algorithms Using the Optimism Principle

1. Design bonuses so that the UCBs are valid: this ensures that with high probability,

$$\text{reg}_t \leq 2b_t(a_t)$$

2. Subject to the above condition, design the bonus to be as tight as possible.

2 Stochastic Linear Contextual Bandits

Stochastic Linear Contextual Bandits extend the multi-armed bandit framework by incorporating context, where each action a at time t depends on an observed context x_t. The expected reward is modeled linearly as:

$$r_t = \langle \theta^*, \phi(x_t, a) \rangle + \epsilon_t$$

The objective is to select actions that maximize cumulative rewards by balancing exploration and exploitation, leveraging the linear relationship between actions and rewards to estimate θ^* and improve decision-making over time.

Setup

Application: Recommendation with personalization.

- Contextual Bandits: we would like to recommend different products to different users – e.g. recommending foods to gourmets and recommending books to researchers, respectively.
- This is different from multi-armed bandits, which aims to Find a single product (action) that maximizes overall user satisfaction.

Protocol

We can write the protol formally as:

1. For $t = 1, 2, \ldots$
2. Observe context $x_t \in X$ (user profile).
3. Takes action $a_t \in A$ (a product).
4. Receive reward: $r_t = f^*(x_t, a_t) + \epsilon_t$
    - f^*: expected reward function.
    - ϵ_t: zero-mean, 1-subgaussian noise. (For concreteness, can think about $\epsilon_t \sim N(0, 1)$.)

Goal

To maximize:

$$\mathbb{E}\left[\sum_{t=1}^{T} r_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} f^*(x_t, a_t)\right]$$

Generalization and Exploration in Learning

In general the algorithms studied so far fall into either the Generalization category, or the Exploration category. Stochastic Linear Regression provides a combination of both, as we can see in the list below:

- SL (Supervised Learning), OL (Online Learning) $\longleftarrow$ **Generalization**
- MAB (Multi-Armed Bandits) $\longleftarrow$ **Exploration**
- SLB (Stochastic Linear Bandits) $\longleftarrow$ **Both Generalization and Exploration**

Linear Realizability and Regret in Contextual Bandits

Assumptions and Definitions

- **Linear Realizability:** $f(x, a) = \langle \phi(x, a), \theta \rangle$, where $\theta \in \mathbb{R}^d$ and $\|\theta\|_2 \leq 1$.
    - $\phi(x, a)$: Known feature transformation.
    - θ^*: Ground truth reward predictor, unknown.
- **Norm Constraint:** $\|\phi(x, a)\|_2 \leq 1$ for all x, a.

Regret Definition

$$\text{Regret}(T) = \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in A} f(x_t, a)\right]}_{\text{optimal policy reward}} - \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} f^*(x_t, a_t)\right]}_{\text{learning agent reward}}$$

MAB as linear contextual bandits

Every MAB problem $(f^*(1), \ldots, f^*(A))$ can be viewed as a linear contextual bandit problem where:

- Dimension $d = A$, where A is the number of actions.
- Each MAB problem can be represented with a constant context $x_t = z_0$ for all t, where z_0 is a dummy context.
- Feature vector $\phi(z_0, a)$ for action a is A-dimensional with all zeros except for a one at the index corresponding to action a.
- The function $f^*(a)$ for any action a can be represented as:

  $f^*(a) = \langle \theta^*, \phi(z_0, a) \rangle$, where $\theta^* = (f^*(1), \ldots, f^*(A))$ representing the true rewards for each action.

  Note: this perspective is useful to keep in mind. This serves as a good 'test case' to assess the sensibility of a linear contextual bandit algorithm. If the algorithm does not even behave sensibly in multi-arm bandit problems (e.g. does not do exploration), then we don't expect it do to well in general.

Algorithm for Stochastic Linear Bandits (SLB)

Objective

- Estimate θ^* (Exploration) and maximize reward (Exploitation) using uncertainty quantification.

Optimism Principle

The optimism principle refers to the fact that we try to presume each candidate can perform better than the average performance we have seen up to time t.

- "world model" = reward model = θ.
- 'plausible world' = set of θs that can plausibly be θ^* and can explain observations $(x_s, a_s, r_s)_{s=1}^{t-1} \longrightarrow$ Construct confidence set for θ^*: $(\Theta_t)_{t=1}^{T}$.
- $\mathbb{P}\left(\forall t, \theta^* \in \Theta_t\right) \geq 1 - \frac{1}{T}$

Algorithm (LinUCB/OFUL)

We can write LinUCB/OFUL algorithm as follows:
For $t = 1, 2, \ldots, T$:

1. Construct Θ_t based on historical data, so that with high probability, $\theta^* \in \Theta_t$.

   For this round t, we will treat this as the only knowledge we have about θ^*: apart from knowing θ^* is in Θ_t, we know nothing further about its location inside Θ_t

2. Observe x_t.

3. For every action a compute the set of plausible values for $f^*(x_t, a)$, defined as

   $$\{\langle \theta, \phi(x_t, a) \rangle : \theta \in \Theta_t\}.$$

   The highest plausible value of $f^*(x_t, a)$ is

   $$\max_{\theta \in \Theta_t} \langle \theta, \phi(x_t, a) \rangle =: UCB_t(x_t, a).$$

4. Take action $a_t = \arg\max_{a \in A} UCB_t(a)$.

Discussion

- Why does it balance exploration and exploitation?

Constructing Confidence Sets and Analyzing LinUCB

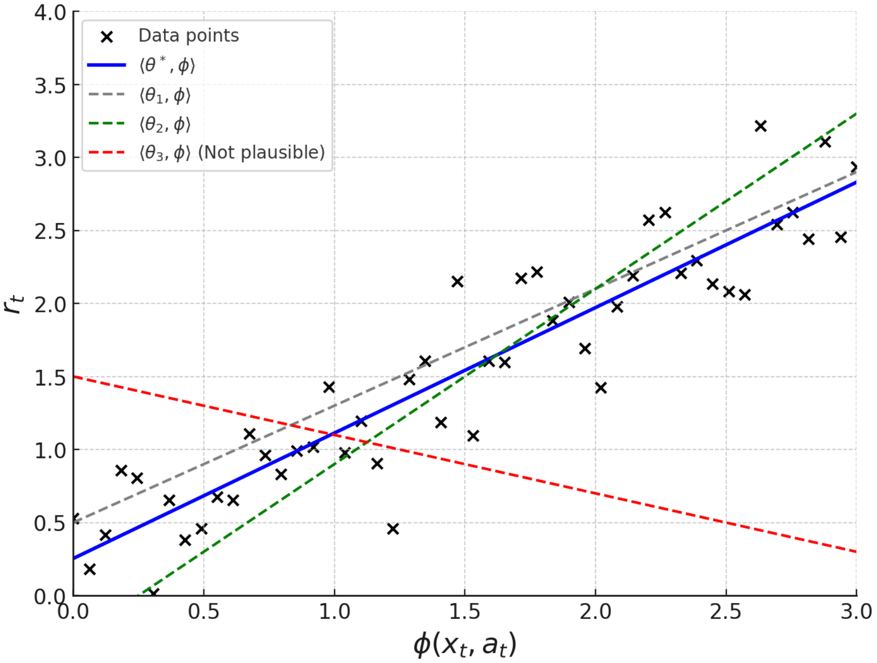

Figure 3: Plot of observed rewards r_t against feature vector values $\phi(x_t, a_t)$ for various actions in the LinUCB algorithm. The blue line represents the true reward predictor using the optimal parameter vector θ^*. The dashed lines indicate alternative linear models based on parameter estimates θ_1 (gray) and θ_2 (green), which are elements of the confidence set Θ_t. Additionally, the red line represents a parameter vector θ_3 that is not plausible for θ^*. The algorithm uses these models to balance exploration and exploitation by estimating different reward functions and selecting actions accordingly.

Questions

1. How to construct Θ_t?
2. How to analyze LinUCB?

Constructing the Confidence Set

Recall that in early lectures, we construct confidence intervals of population mean by first choosing its center as the sample mean, and choosing their widths appropriately using concentration inequalities such as Hoeffding's inequality. Now we are faced with a similar task: construct confidence set for $\theta^* \in \mathbb{R}^d$, a multi-dimensional vector. This motivates us to use a similar two-step procedure: (1) determine the center of the confidence set, (2) quantify the proximity of θ^* to the confidence set center.

- Constructing the center: "best guess" of θ^* based on data, as can be seen in Figure 3.

- The standard solution that can be used is least squares:

$$\hat{\theta}^t(\lambda) = \underset{\theta}{\operatorname{argmin}} \sum_{s=1}^{t-1} \left(\langle \theta, \phi(x_s, a_s)\rangle - r_s\right)^2 + \lambda \|\theta\|_2^2$$

  Here, ℓ_2-regularization is added. This is mainly to ensure the uniqueness of the solution. For the rest of the lecture, we will mainly focus on $\lambda = 1$.

- In general, we cannot hope for pointwise closeness of $\hat{\theta}(t)$ to θ^* due to potential data degeneracy.

Problem in Data Degeneracy

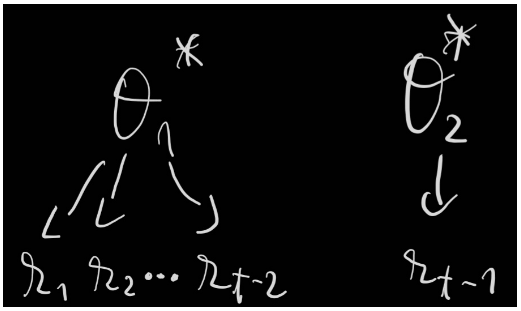

Figure 4: Illustration of the dependency structure between the parameter vectors θ_1^* and θ_2^* and their associated observations. The left side shows θ_1^* influencing a series of observed rewards $r_1, r_2, \ldots, r_{t-2}$, indicating that θ_1^* can be informed by multiple observations. In contrast, θ_2^* on the right only influences a single observation r_{t-1}, highlighting a lack of information and resulting in an under-determined system. This limited observation structure provides sufficient data to estimate θ_1^* but leaves θ_2^* inadequately constrained, thus requiring diverse feature vectors for reliable estimation of both parameters.

An example of data degeneracy is for $d = 2$, $\theta^* = \begin{pmatrix}\theta_1^* \\ \theta_2^*\end{pmatrix}$. Suppose the features of the context-actions seen at the beginning of round t is: $\phi_1 = \phi_2 = ... = \phi_{t-2} = \begin{pmatrix}1 \\ 0\end{pmatrix}$ are identical. In this case we also have $\phi_{t-1} = \begin{pmatrix}0 \\ 1\end{pmatrix}$.

causing issues in determining the values.

For this case we can write the first reward function as $r_1 = \langle \theta^*, \phi_t \rangle + \epsilon_t = \theta_1^* + \epsilon_1$.

So then again in this specific case we will have the following dependencies for θ_1^*:

- $\theta_1^* \implies r_1$
- $\theta_1^* \implies r_2$
- $\vdots$
- $\theta_1^* \implies r_{t-2}$

And then for θ_2^* we only have $\theta_2^* \implies r_{t-1}$, which is a significant problem as the feature vectors only provide information about θ_1^* through repeated identical observations, but not enough data for θ_2^*. Since θ_2^* is only associated with one observation, the estimate for θ_2^* is highly unreliable and heavily influenced by noise. This can also be seen in Figure 4. Consequently, this lack of unique determination results in an under-determined system that cannot accurately estimate both parameters. Diverse feature vectors are needed to span the parameter space and ensure each parameter is adequately observed, enabling reliable estimation of both θ_1^* and θ_2^*.

In this specific case we can also deduce that $|\hat{\theta}_1^t - \theta_1^*|$ is small while $|\hat{\theta}_2^t - \theta_2^*|$ is large.

The construction of Θ_t depends critically on historical ϕ_s values, which highlights the importance of data quality and integrity in determining the effectiveness of the LinUCB algorithm.