

Multi-Armed Bandits

Reading

- Sutton, Richard S., and Barto, Andrew G. Reinforcement learning: An introduction. MIT press, 2018.
 - <http://www.incompleteideas.net/book/the-book-2nd.html>
 - Chapter 2
- Slivkins, Aleksandrs. "Introduction to multi-armed bandits." Foundations and Trends® in Machine Learning 12.1-2 (2019)
 - <https://arxiv.org/pdf/1904.07272>
 - Chapter 1
- Agarwal, Alekh, et al. "Reinforcement learning: Theory and algorithms." *CS Dept., UW, WA, USA, Tech. Rep 32* (2019): 96.
 - https://rltheorybook.github.io/rltheorybook_AJKS.pdf
 - Chapter 6 (will consider a modified version of their proof)

Overview

- Multi-armed bandits have many applications
 - Dynamic advertising on websites
 - Dynamic pricing
 - Investment
 - etc.
- It's a sequential decision-making problem
 - Simpler than the general RL setting since there is no state
 - Pick one out of k actions at each step so as to maximize some reward (e.g., profits)
- Agenda
 - formalize the standard multi-armed bandit setting
 - derive the popular upper confidence bound algorithm

Multi-Armed Bandit

- Suppose you are in a casino all by yourself
 - There are k slot machines, each with a different probability of success
 - At any given time, you can only be on one slot machine
 - Pull one arm
 - You would like to learn which slot machine is the best

Multi-armed Bandits Formalization

- At each time, you can select 1 out of k actions
- Each action has an unknown expected reward
 - E.g., slot machine payout rate
- Your goal is to learn the expected rewards over time
 - Then you select the action with highest expected reward
- Bonus points if you can minimize the number of attempts
 - At the beginning, you are in *exploration* phase
 - Trying different actions randomly and seeing the rewards
 - Eventually, you switch to the *exploitation* phase
 - When you have a good estimate of rewards, you pick actions to maximize the rewards
 - Balancing the 2 is one of the fundamental challenges in RL

Multi-armed Bandits Formalization, cont'd

- The agent has K possible actions, i.e., $A = \{a_1, \dots, a_K\}$
- At each round t , you taken an action A_t and observe a reward R_{t+1}
- Each action a has an unknown reward function is
$$R_e(a) = \mathbb{E}[R_{t+1} | A_t = a]$$
 - where A_t is the random variable for the action at time step t
 - where R_{t+1} is the random variable for the reward at time step $t + 1$
 - By convention, the reward is received one step after the action is taken
- Goal: estimate $R_e(a)$ for all actions a and learn the best action
$$a^* = \underset{a}{\operatorname{argmax}} R_e(a)$$

Naïve Approach

- How do we estimate the expected reward of each action?
 - Hint: what probabilistic tools did we discuss?
- Try each action N number of times and collect the rewards
 - Calculate the average reward per action
 - As $N \rightarrow \infty$, the average will converge to the true expected reward (law of large numbers)
- What can we say about a specific finite N ?
 - For any N , can construct a confidence interval around your current estimate
 - E.g., using a concentration bound like Hoeffding's inequality

Probability Aside: Hoeffding's Inequality

- Let $X_1, \dots, X_n$ be n independent random variables
 - Each bounded by $a_i \leq X_i \leq b_i$
- Let $S_n = X_1 + \dots + X_n$
- Hoeffding's Theorem:

$$\mathbb{P}[S_n - \mathbb{E}[S_n] \geq t] \leq \exp \left\{ -\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right\}$$

- Given a sample S_n , bound its deviation from the true mean
- The larger t is, the higher the probability the mean is within t of the sample
- The smaller the bounds $(b_i - a_i)$, the tighter the bound on S_n
- A type of concentration bound

Probability Aside: Hoeffding's Inequality, cont'd

- Hoeffding's Theorem:

$$\mathbb{P}[S_n - \mathbb{E}[S_n] \geq t] \leq \exp \left\{ -\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right\}$$

- Two-tailed version:

$$\mathbb{P}[|S_n - \mathbb{E}[S_n]| \geq t] \leq 2 \exp \left\{ -\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2} \right\}$$

– Proved using the union bound

Probability Aside: Hoeffding's Inequality, cont'd

- Suppose we know the reward varies by at most some $B > 0$
 - For simplicity, suppose $B = 1$

- The bound simplifies to:

$$\mathbb{P}[S_n - \mathbb{E}[S_n] \geq t] \leq \exp\left\{-\frac{2t^2}{n}\right\}$$

- Furthermore, suppose we are interested in bounding the mean

$$\begin{aligned}\mathbb{P}\left[\frac{1}{n}(S_n - \mathbb{E}[S_n]) \geq t\right] &= \\ &= \mathbb{P}[S_n - \mathbb{E}[S_n] \geq nt] \leq \exp\{-2t^2n\}\end{aligned}$$

Probability Aside: Hoeffding's Inequality, cont'd

$$\mathbb{P} \left[\frac{1}{n} |S_n - \mathbb{E}[S_n]| \geq t \right] \leq 2\exp\{-2t^2n\}$$

- How do we construct a 95%-confidence interval around $\frac{S_n}{n}$?
 - Solve for t such that

$$2\exp\{-2t^2n\} = 0.05$$

$$\text{i.e., } t = c \sqrt{\frac{1}{n}}$$

$$\text{– where } c = \sqrt{-0.5 * \log(0.05/2)} = \sqrt{0.5 * \log(2/0.05)}$$

- In general, for any confidence $1 - \delta$:

$$c = \sqrt{0.5 * \log(2/\delta)}$$

Probability Aside: Hoeffding's Inequality, cont'd

$$\mathbb{P} \left[\frac{1}{n} |S_n - \mathbb{E}[S_n]| \geq t \right] \leq 2\exp\{-2t^2n\}$$

- How do we construct a $1 - \delta$ -confidence interval around $\frac{S_n}{n}$?

- Set $t = c \sqrt{\frac{1}{n}}$

- where $c = \sqrt{0.5 * \log(2/\delta)}$

- So finally:

$$\mathbb{P} \left[\frac{1}{n} |S_n - \mathbb{E}[S_n]| \geq c \sqrt{\frac{1}{n}} \right] \leq \delta$$

Probability Aside: Hoeffding's Inequality, cont'd

- So finally,

$$\mathbb{P} \left[\frac{1}{n} |S_n - \mathbb{E}[S_n]| \geq c \sqrt{\frac{1}{n}} \right] \leq \delta$$

- The $1 - \delta$ confidence interval for the true $\mathbb{E} \left[\frac{S_n}{n} \right]$ is thus

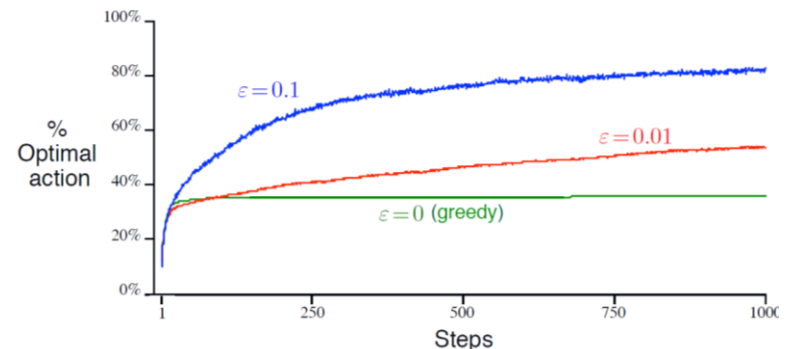
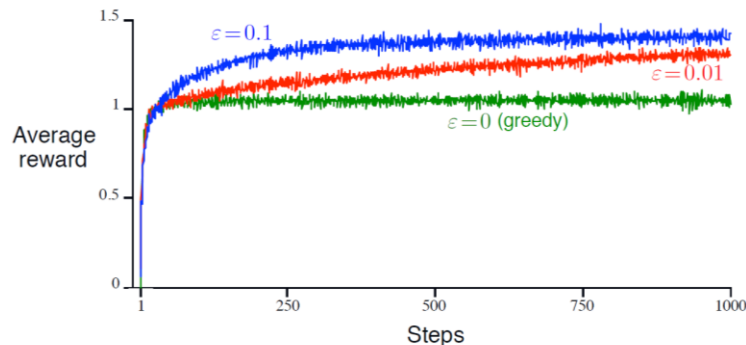
$$\left[\frac{S_n}{n} - c \sqrt{\frac{1}{n}}, \frac{S_n}{n} + c \sqrt{\frac{1}{n}} \right]$$

Multi-armed Bandits, Online Version

- In practice, we have to choose an action every time
 - Can't pre-collect N datapoints for each action
- So how do we choose that action?
 - Keep a running average of each action
 - At each step, choose the action with the highest average
- This is OK, but has a major limitation
 - Some actions may get very few points if they get a few bad samples
 - You are not guaranteed to find the best action in the limit
 - How do we fix this issue?

Multi-armed Bandits, ϵ -greedy

- ϵ -greedy action selection
- At each step, choose the action with the highest average
 - But with probability $1 - \epsilon$, for small $\epsilon > 0$
 - With probability ϵ , pick another action at random
 - $k - 1$ other actions, so other actions get $\frac{\epsilon}{k-1}$ probability each
- 10-armed example from the book
 - The $\epsilon = 0.1$ case converges fastest in this example
 - The $\epsilon = 0$ case eventually plateaus



Multi-armed Bandits, ϵ -greedy Implementation

- When computing the running average, we don't need to add up all past rewards every time
 - E.g., suppose average at time t is $q^t(a) = \frac{1}{t} \sum_{i=1}^t R_i$
 - For simplicity, here we assume we pull a each time
 - What is the average $q^{t+1}(a)$, in terms of $q^t(a)$ and R_{t+1} ?

$$\begin{aligned} q^{t+1}(a) &= \frac{R_1 + R_2 + \dots + R_{t+1}}{t+1} \\ &= \frac{1}{t+1} (R_1 + \dots + R_t) + \frac{1}{t+1} R_{t+1} \\ &= \frac{t}{t+1} \frac{(R_1 + \dots + R_t)}{t} + \frac{1}{t+1} R_{t+1} \\ &= \frac{t}{t+1} q^t(a) + \frac{1}{t+1} R_{t+1} \\ &= q^t(a) + \frac{1}{t+1} (R_{t+1} - q^t(a)) \end{aligned}$$

Multi-armed Bandits, ϵ -greedy Implementation, cont'd

- When computing the running average, we don't need to add up all past rewards every time
 - E.g., suppose average at time t is $q^t(a) = \frac{1}{t} \sum_{i=1}^t R_i$
 - What is the average $q^{t+1}(a)$, in terms of $q^t(a)$ and R_{t+1} ?
$$q^{t+1}(a) = q^t(a) + \frac{1}{t+1} (R_{t+1} - q^t(a))$$
- Compute the difference between the new reward and the running average
 - This is a simple example of temporal difference learning (more later)

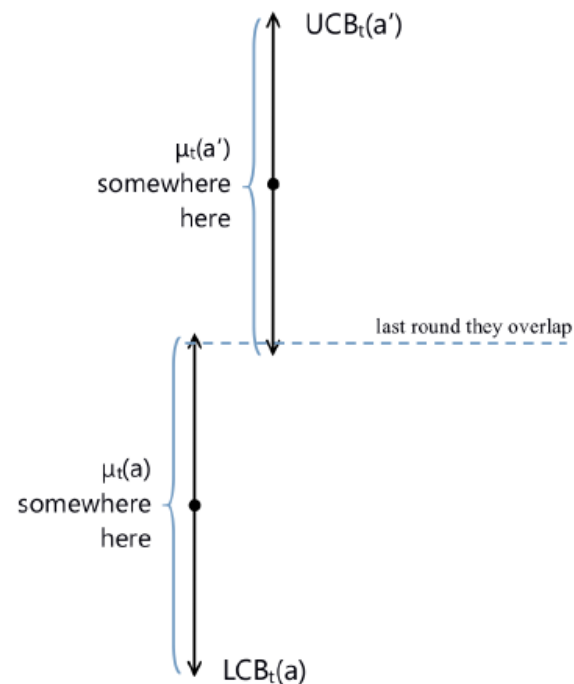
Successive Elimination Algorithm

- Can you design an algorithm that declares victory with high probability?
 - i.e., it keeps trying actions until it is 95%-confident that it has identified the best action
- For simplicity, suppose we have 2 actions
 - Suppose we keep running 95%-confidence intervals for each $[LCB(a_1), UCB(a_1)], [LCB(a_2), UCB(a_2)]$
- Can terminate algorithm when one interval is entirely larger than the other, e.g.,:

$$LCB(a_2) > UCB(a_1)$$

Successive Elimination Algorithm, cont'd

- What about more actions?
 - Successively eliminate actions whose upper bound is lower than the best action's lower bound



Calculating Confidence Intervals Online

- Can't directly apply Hoeffding's inequality to calculate the confidence intervals
 - Why?
 - The rewards R_t are not necessarily independent!
 - Consider the following algorithm:
 - sample action a_1 two times and then only sample a_1 a 3rd time if $R_1 = R_2 = 0$
 - Clearly R_3 only exists when $R_1 = R_2 = 0$
 - How do we get around this issue?

Calculating Confidence Intervals Online, cont'd

- Suppose we are allowed to make a total of T actions
- Each action a gets a total of $0 < n(a) \leq T$ actions
 - Note that $n(a)$ is random and depends on the algorithm
 - Let $S_{n,a}$ be the sample sum for the rewards received when taking action a
- The following bound holds for any algorithm

$$\mathbb{P} \left[\frac{1}{n(a)} |S_{n,a} - \mathbb{E}[S_{n,a}]| \geq \frac{\sqrt{2T \log(1/\delta)}}{n(a)} \right] \leq \delta$$

- Proof requires theory of martingales
 - Shown at the end of this deck

Calculating Confidence Intervals Online, cont'd

- The following bound holds for any algorithm

$$\mathbb{P} \left[\frac{1}{n(a)} |S_{n,a} - \mathbb{E}[S_{n,a}]| \geq \frac{\sqrt{2T \log(1/\delta)}}{n(a)} \right] \leq \delta$$

- Very similar to the original Hoeffding bound

– If we assume $T \approx n(a)$, we get

$$\mathbb{P} \left[\frac{1}{n(a)} |S_{n,a} - \mathbb{E}[S_{n,a}]| \geq \sqrt{\frac{2 \log \left(\frac{1}{\delta} \right)}{n(a)}} \right] \leq \delta$$

– Similar to original Hoeffding bound except $n(a)$ is random

- Ultimately the bounds are the same: T is fixed, so the choice of δ will determine the confidence interval size

Upper-Confidence-Bound Algorithm

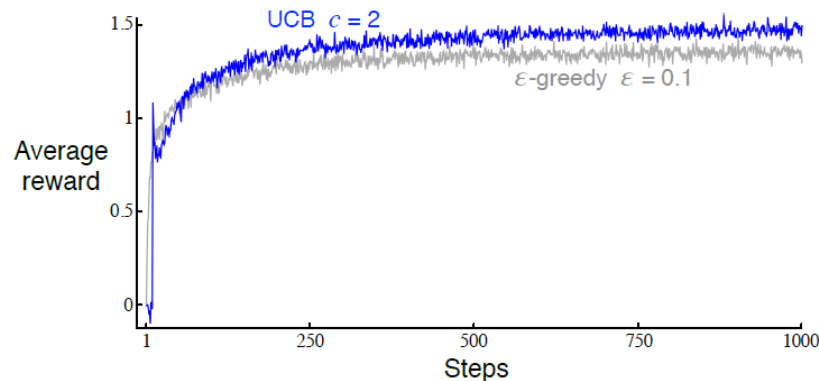
- We know that without exploration we are almost certainly going to converge to a suboptimal action
- On the other hand, too much exploration may take a long time to converge
 - So far, we've seen ϵ -greedy exploration, which indiscriminately selects the next action randomly
- Is there a way to perform targeted exploration?
- Pick the action with the highest UCB!
 - Why is this a good idea?
 - Either the highest-UCB action is already the best
 - or the highest-UCB action has a large confidence interval, which means it could benefit from more exploration
 - In both cases it makes sense to select that action

Upper-Confidence-Bound Algorithm, cont'd

- Suppose we have a choice of K actions
- For $t \in [1, K]$:
 - Take each action once and observe the reward
- For $t > K$:
 - Calculate running reward averages $q^t(a_i)$ for each action a_i
 - Take action $a_t = a_{i^*}$, where $i^* = \underset{i}{\operatorname{argmax}} q^t(a_i) + \sqrt{\frac{c}{n(a_i)}}$
 - where $c = 2 \log\left(\frac{1}{\delta}\right)$
 - Book uses a different c (no time to prove)
 - » Tighter confidence bounds can be derived specific to UCB
 - Observe corresponding reward r_t
 - Update $q^t(a_{i^*})$ and increment $n(a_{i^*})$

Upper-Confidence-Bound Algorithm, cont'd

- UCB algorithm generally outperforms ϵ -greedy
- UCB is not widely used in the general RL setting, however
 - May introduce a lot of variance in a high-dimensional action space
 - If we have many actions, we will require a lot of data in order to try all actions enough times and get good confidence intervals



Conclusion

- Multi-armed bandits is a well-studied setting with a number of strong theoretical results
- It can be considered as a simplified RL setting where the environment has no state
- In this lecture, we considered the case where we made no assumptions about rewards
 - Except that they are bounded
- Next time, we'll look at Bayesian bandits where we assume a prior about the reward distribution

Proof of Confidence Bound

- Suppose we are allowed to make a total of T actions
- Each action a gets a total of $0 < n(a) \leq T$ actions
 - Note that $n(a)$ is random and depends on the algorithm
 - Let $S_{n,a}$ be the sample average for the rewards received when taking action a
- The following bound holds for any algorithm

$$\mathbb{P} \left[\frac{1}{n(a)} |S_{n,a} - \mathbb{E}[S_{n,a}]| \geq \frac{\sqrt{2T \log(1/\delta)}}{n(a)} \right] \leq \delta$$

- Sutton book uses a slightly different bound but

Probability Aside: Hoeffding-Azuma Inequality

- **Definition:** A sequence of random variables $X_0, \dots, X_T$ is a martingale difference sequence if

$$\begin{aligned}\mathbb{E}[X_t] &< \infty \\ \mathbb{E}[X_t | X_0, \dots, X_{t-1}] &= 0\end{aligned}$$

- **Theorem [Hoeffding-Azuma Inequality]:** Let $X_0, \dots, X_T$ be a martingale difference sequence and suppose $|X_t - X_{t-1}| \leq c_t$. Then, for all $\epsilon > 0, T > 0$:

$$\mathbb{P}\left[\sum_{i=0}^T X_i \geq \epsilon\right] \leq \exp\left(\frac{-\epsilon^2}{2 \sum_{i=1}^T c_i^2}\right)$$

Proof of Confidence Bound, cont'd

- Consider a fixed action a and fixed algorithm $\mathcal{A}$
 - Let $\hat{\mu}_a^t = \frac{S_{n_t,a}}{n_t(a)}$ be the running average at time step t
 - Let $\mu_a = \mathbb{E}[R_{t+1}|A_t = a]$ be the true expected reward for action a
 - Assume each action a is tried once initially, with random reward R_a
- Define the following random variables
$$X_0 = R_a - \mu_a, X_1 = \mathbf{1}\{A_1 = a\}(R_2 - \mu_a), \dots, X_T = \mathbf{1}\{A_T = a\}(R_{T+1} - \mu_a)$$
 - Each X_t is 0 when action a is not taken at time t and r_t otherwise (normalized to be 0-mean by subtracting μ_a)

Proof of Confidence Bound, cont'd

- Define the following random variables
$$X_0 = R_a - \mu_a, X_1 = \mathbf{1}\{A_1 = a\}(R_2 - \mu_a), \dots, X_T = \mathbf{1}\{A_T = a\}(R_{T+1} - \mu_a)$$
 - Each X_t is 0 when action a is not taken at time t and r_t otherwise (normalized to be 0-mean by subtracting μ_a)
- Notice that $\mathbb{E}[X_t | X_1, \dots, X_{t-1}] = 0$
 - Given all history, $\mathbf{1}\{A_1 = a\}$ is deterministic
 - Decided by the algorithm $\mathcal{A}$
 - Either $\mathbf{1}\{A_1 = a\} = 0$ (in which case expectation is 0)
 - Or $\mathbf{1}\{A_1 = a\} = 1$, in which case $\mathbb{E}[X_t | X_1, \dots, X_{t-1}] = \mathbb{E}[R_t - \mu_a] = 0$
- Thus, $X_1, \dots, X_T$ is a martingale difference sequence

Proof of Confidence Bound, cont'd

- Define the following random variables

$$X_0 = R_a - \mu_a, X_1 = \mathbf{1}\{A_1 = a\}(R_2 - \mu_a), \dots, X_T = \mathbf{1}\{A_T = a\}(R_{T+1} - \mu_a)$$

– Each X_t is 0 when action a is not taken at time t and r_t otherwise (normalized to be 0-mean by subtracting μ_a)

- Also notice that $|X_t - X_{t-1}| \leq 1$ for all t

– Recall $R_t \in [0,1]$, which means $\mu_a \in [0,1]$ and hence

$$R_t - \mu_a \in [0,1] \text{ and } X_t \in [0,1]$$

- By the Hoeffding-Azuma inequality, for any t

$$\mathbb{P} \left[\sum_{i=0}^t X_i \geq \epsilon \right] \leq \exp \left(\frac{-\epsilon^2}{2 \sum_{i=1}^t 1^2} \right) \leq \exp \left(\frac{-\epsilon^2}{2t} \right)$$

Proof of Confidence Bound, cont'd

- Define the following random variables

$$X_0 = R_a - \mu_a, X_1 = \mathbf{1}\{A_1 = a\}(R_2 - \mu_a), \dots, X_T = \mathbf{1}\{A_T = a\}(R_{T+1} - \mu_a)$$

- By the Hoeffding-Azuma inequality, for any fixed t

$$\mathbb{P}\left[\sum_{i=0}^t X_i \geq \epsilon\right] \leq \exp\left(\frac{-\epsilon^2}{2 \sum_{i=0}^t 1^2}\right) \leq \exp\left(\frac{-\epsilon^2}{2t}\right)$$

- Notice that

$$\begin{aligned}\sum_{i=0}^t X_i &= \sum_{i=0}^t \mathbf{1}\{A_i = a\}R_{i+1} - \sum_{i=0}^t \mathbf{1}\{A_i = a\}\mu_a \\ &= S_{n_t, a} - n_t(a)\mu_a \\ &= n_t(a)\hat{\mu}_a^t - n_t(a)\mu_a\end{aligned}$$

Proof of Confidence Bound, cont'd

- Define the following random variables

$$X_0 = R_a - \mu_a, X_1 = \mathbf{1}\{A_1 = a\}(R_2 - \mu_a), \dots, X_T = \mathbf{1}\{A_T = a\}(R_{T+1} - \mu_a)$$

- By the Hoeffding-Azuma inequality, for any fixed t

$$\mathbb{P}\left[\sum_{i=0}^t X_i \geq \epsilon\right] \leq \exp\left(\frac{-\epsilon^2}{2 \sum_{i=0}^t 1^2}\right) \leq \exp\left(\frac{-\epsilon^2}{2t}\right)$$

- Finally,

$$\begin{aligned}\left[\sum_{i=0}^t X_i \geq \epsilon\right] &= \mathbb{P}[n_t(a)\hat{\mu}_a^t - n_t(a)\mu_a \geq \epsilon] \\ &= \mathbb{P}\left[\hat{\mu}_a^t - \mu_a \geq \frac{\epsilon}{n_t(a)}\right] \leq \exp\left(\frac{-\epsilon^2}{2t}\right)\end{aligned}$$

Proof of Confidence Bound, cont'd

$$\mathbb{P} \left[\hat{\mu}_a^t - \mu_a \geq \frac{\epsilon}{n_t(a)} \right] \leq \exp \left(\frac{-\epsilon^2}{2t} \right)$$

- Solving for $\delta = \exp \left(\frac{-\epsilon^2}{2t} \right)$, we get

$$\epsilon = \sqrt{-2t \log(\delta)} = \sqrt{2t \log(1/\delta)}$$

- Thus, for $1 - \delta$ confidence:

$$\mathbb{P} \left[\hat{\mu}_a^t - \mu_a \geq \frac{\sqrt{2t \log(1/\delta)}}{n_t(a)} \right] \leq \delta$$

– Plugging in $T = t$, we get the final result