A Gentle Introduction of Multi-Armed Bandit

Qingyun Wu

11/15/2021

What is Multi-Armed Bandit?

If you have done your homework…

The ***multi-armed bandit*** *problem is a problem in which a fixed limited set of resources must be allocated between competing (alternative) choices in a way that maximize their expected gain, when each choice's probabilities are only partially known at the time of allocation.* *--Wiki [Multi-armed bandit]*

What is Multi-Armed Bandit?

Environment
(the guy who gives the player payoff)

competing choices in the environment, aka, **arms**

Player
(the guy who allocate the resource)

*The **multi-armed bandit** problem is a problem in which a fixed limited set of resources must be allocated between competing (alternative) choices in a way that maximize their expected gain, when each choice's probabilities are only partially known at the time of allocation.* *--Wiki [Multi-armed bandit]*

- A two-party game: the player and the world/environment
- What the player need to do: sequentially allocate resources to each arm
- The rules:
  - Every time an arm is played, it generate reward with certain probabilities (the player does know it a prior)
  - The player needs to pay (i.e., allocate resource) to play, and only have limited total budget to play
- Goal of the player: maximize the expected gain/reward

Question: What strategy the player should use to achieve his/her goal?

How will you play if you are the player? (A test of human intelligence ☺)

- Go play at https://axyyu.github.io/multi-armed-bandit/

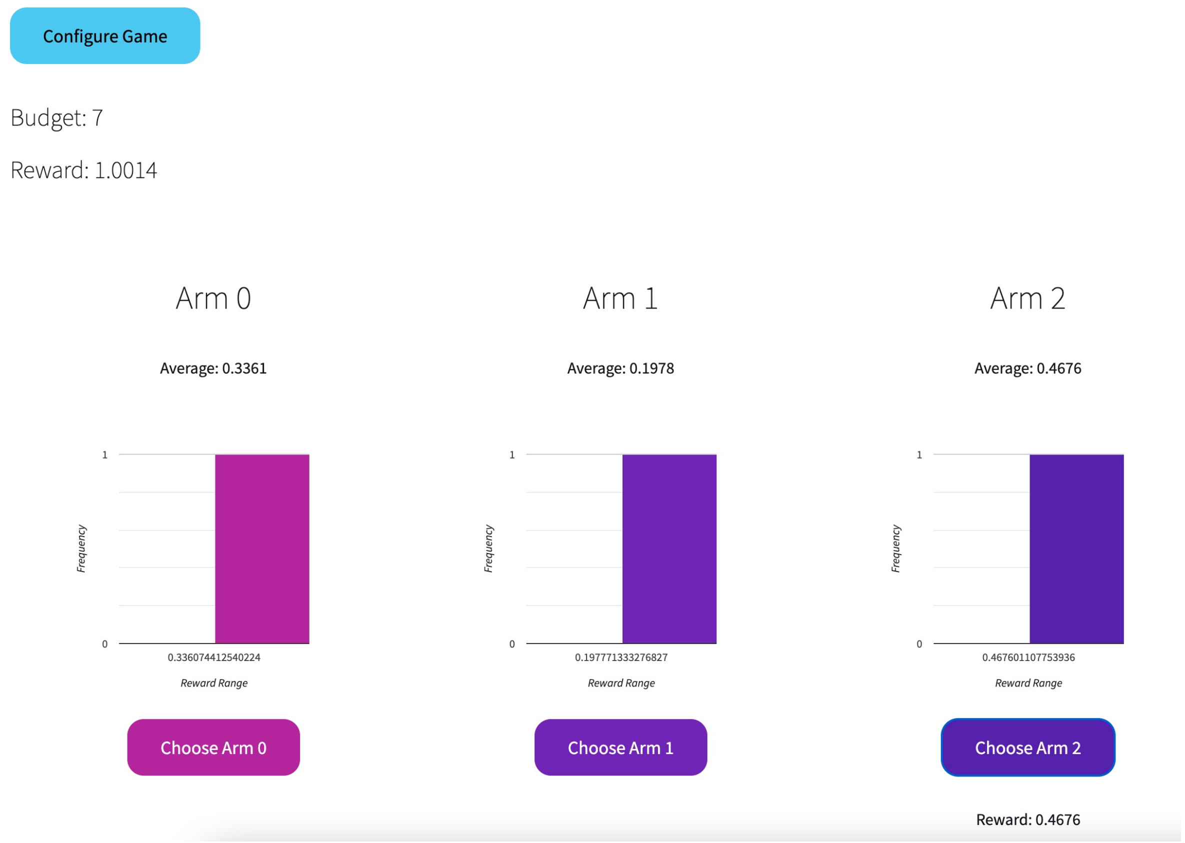
Configure Game

Budget: 10 — How many times you can pull an arm

Reward: 0.0000 — How to maximize this?

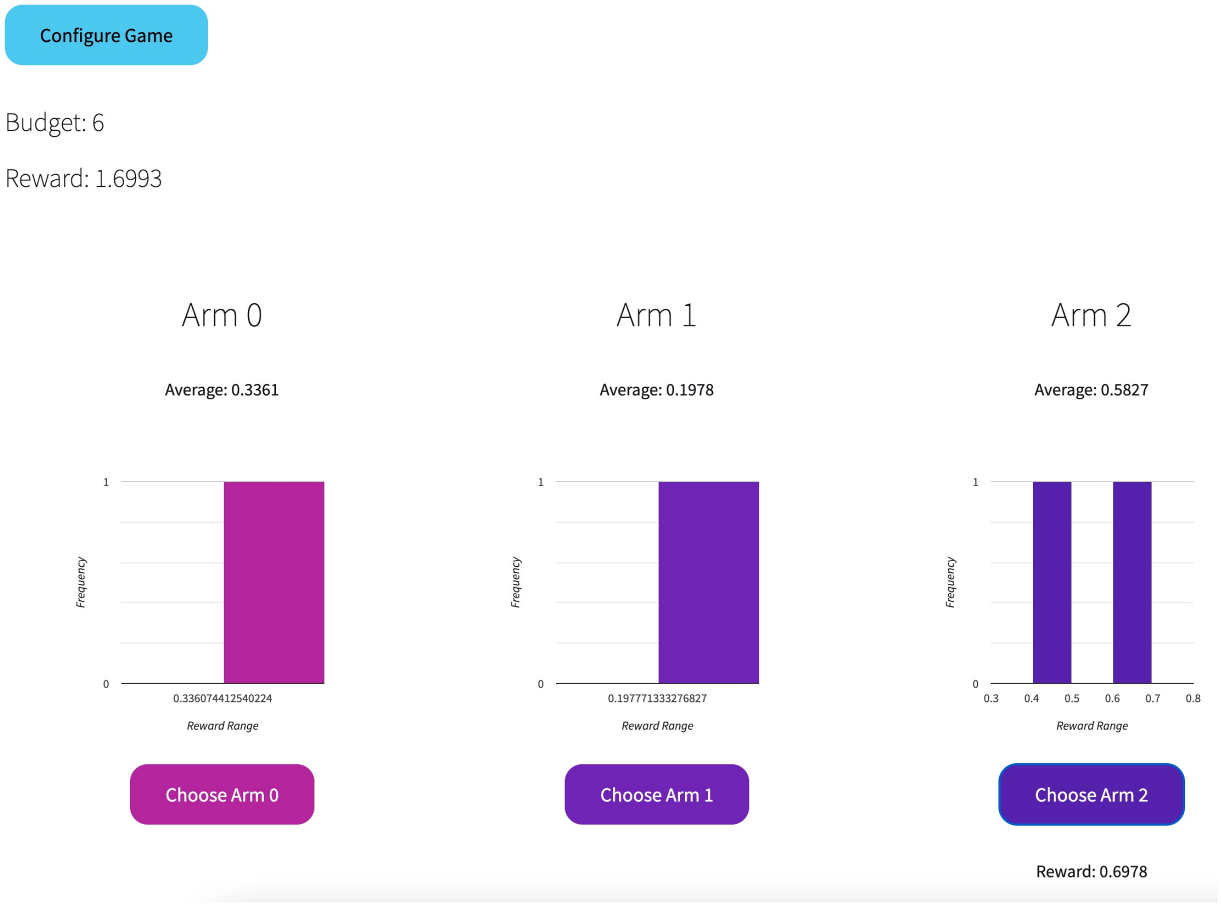
Arm 0	Arm 1	Arm 2
Average: N/A	Average: N/A	Average: N/A
Select this arm once to view a histogram.	Select this arm once to view a histogram.	Select this arm once to view a histogram.
Choose Arm 0	Choose Arm 1	Choose Arm 2

Exploration vs Exploitation Dilemma



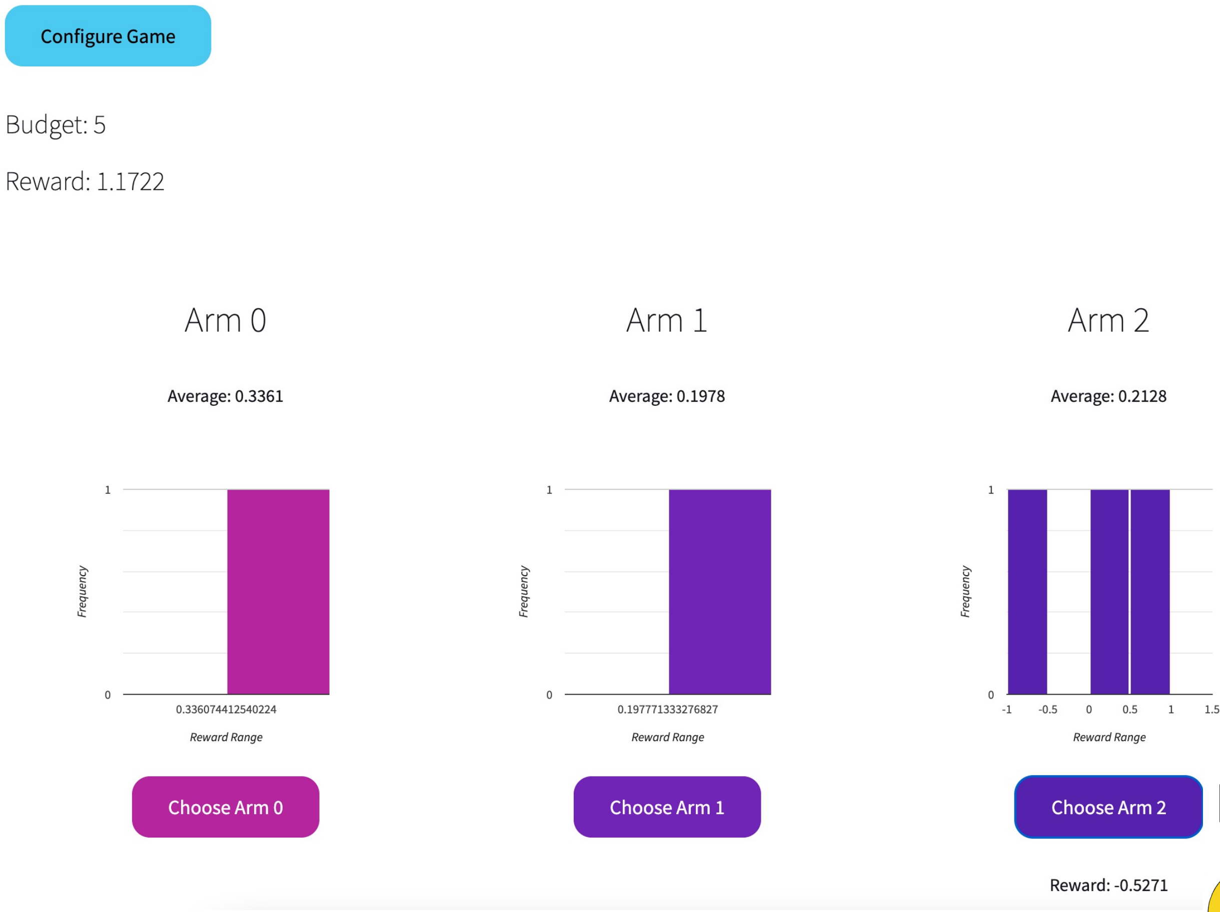
- Exploration: try every choice (uniformly)

Exploration vs Exploitation Dilemma



- Exploration: try every choice (uniformly)
- Exploitation: spend resource on a particular one (based on your knowledge)

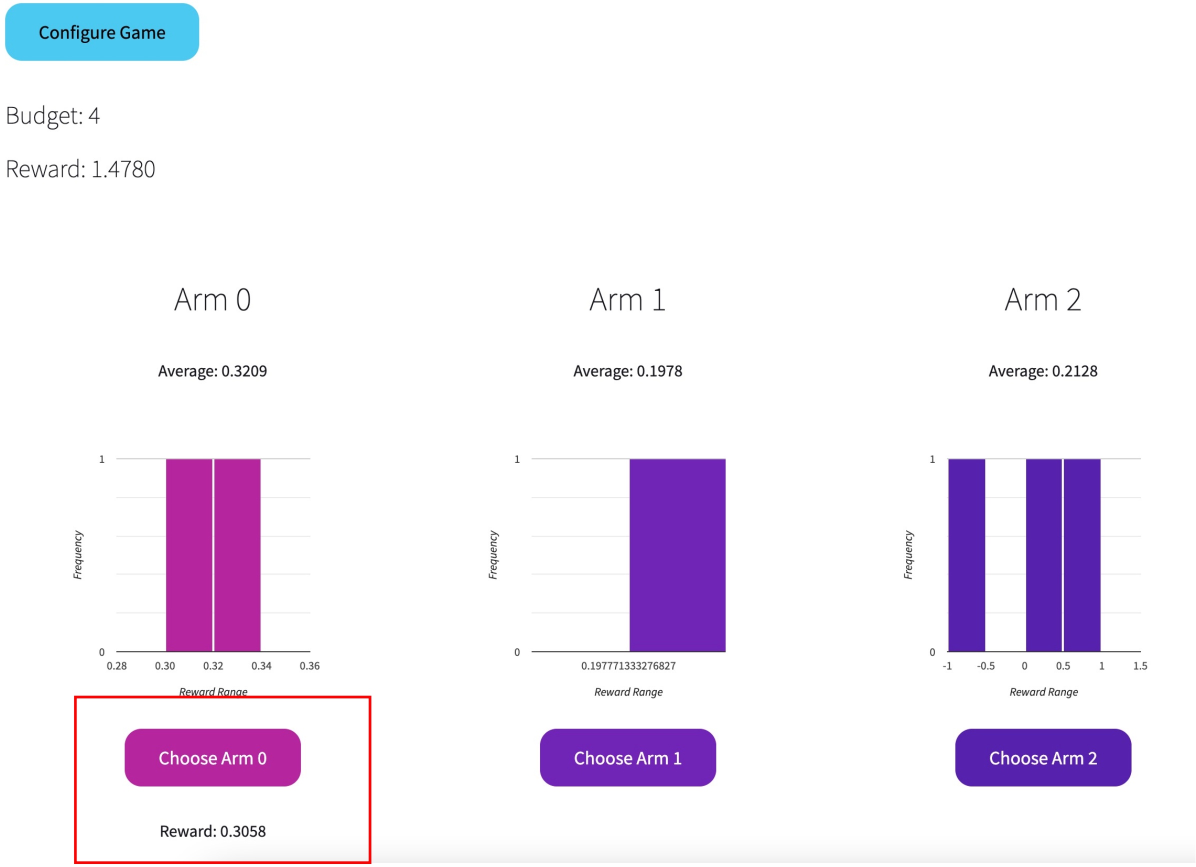
Exploration vs Exploitation Dilemma

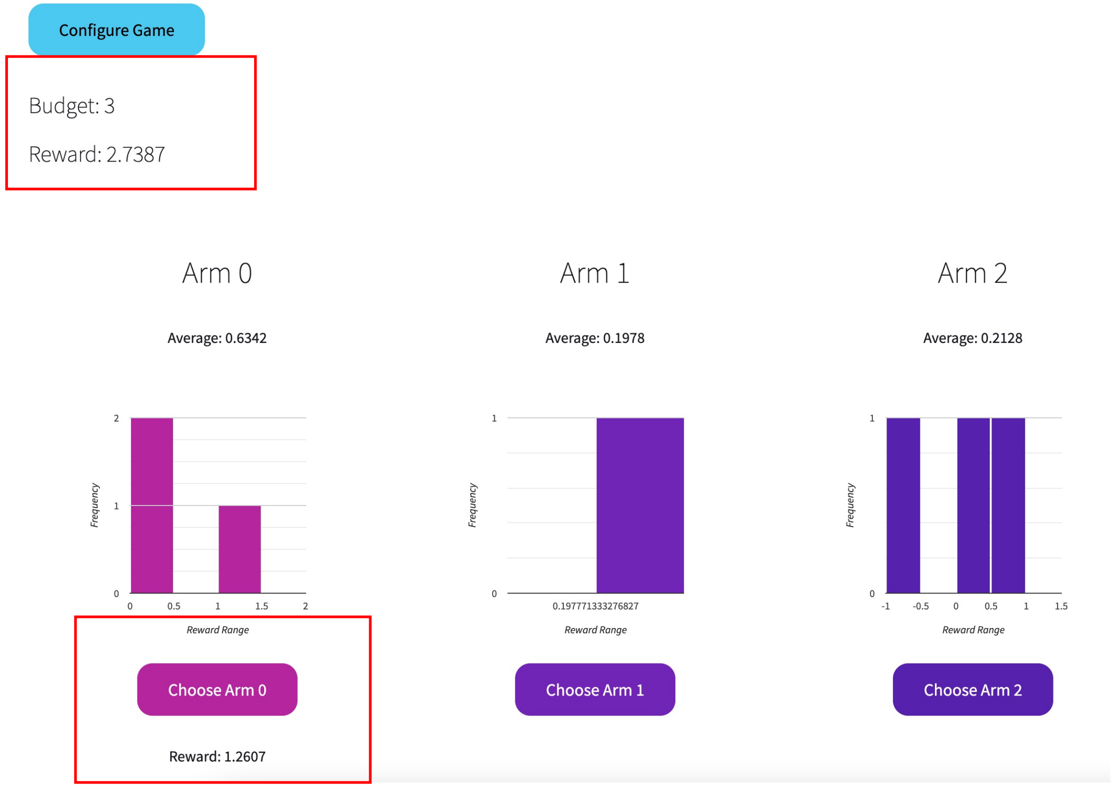


- Exploration: try every choice (uniformly)
- Exploitation: spend resource on a particular one (based on your knowledge)

Exploration vs Exploitation Dilemma

lesson learned....





Reflections

- How should we better play the game?
  - Exploration is in general needed (especially at the beginning of the game).
  - Need to do some kind of exploitation.
    - But it is subtle when should we start this and to what extend should we trust our knowledge about the world.

[Hint] A key element in the decision-making processing: Uncertainty.

Intuitively, it is good to explore more when we are very uncertain about the goodness of the choices, and exploit when we are more certain.

Key questions:

1. How to measure uncertainty?
2. How to use the uncertainty?

Notations:
Arms: *0, 1, 2, 3, .., K-1*
Total budget: *N*
You pay 1 for each play

Algorithm 0

- At round $\mathrm{i} = 0, \ldots, K - 1$: Play arm i — exploration
- At round $\mathrm{i} = K, \ldots, N$
  - Play the arm with the highest average reward — exploitation

- Measurement of the uncertainty: whether an arm is played or not (binary).
- How to use the uncertainty: explore when there are uncertainty, and exploit when there is uncertainty.

Although the way to use the uncertainty seems reasonable, the uncertainty measurement is problematic.
(Think about WHY it is problematic.)

Algorithm 1: ε-greedy

Set ε = 0.1

- At round $\mathrm{i} = 0, \dots, N$ exploration
  - With probability ε, play an arm uniformly at random; and with probability 1- ε, play the arm with the highest average reward (break tie randomly) exploitation

- Measurement of the uncertainty: ε (a probability)
- How it is using intuition (i.e.,) : random sampling according to the uncertainty level.

1. The uncertainty level is fixed to be ε. Is this good enough?
2. How do know the value of ε ?

Algorithm 2: (adaptive) ε-greedy

Set ε = 1.0

- At round $\mathrm{i} = 0, \ldots\ N$
  - With probability $\varepsilon/\sqrt{i+1}$, play an arm uniformly at random; and with probability 1- $\varepsilon/\sqrt{i+1}$, play the arm with the highest average reward

exploration

exploitation

- Measurement of the uncertainty: whether an arm is played, and then $\varepsilon/\sqrt{i+1}$ (a probability)
- How it is using intuition (i.e.,) : random sampling according to the uncertainty level.

The uncertainty level is $\varepsilon/\sqrt{i+1}$.

Good enough?

Another category of algorithms: UCB

- UCB: Upper Confidence Bound
- Key idea of the UCB-style algorithms:
  - Maintain an **estimate on each arm's reward** (according to our historical observations): $\hat{r}_{a,i}$ for each arm *a* at iteration *i*
  - Maintain a **confidence** level on our estimation $B_{a,i}$
  - Action strategy: optimism in the face of uncertainty

Being optimistic in the face of uncertainty ☺ $\boxed{\arg\max_{\{a=0,..K-1\}}}\ \boxed{\hat{r}_{a,i} + B_{a,i}}$ Upper confidence bound of the reward

Well, it seems the confidence level reflects the uncertainty. But how do we get the confidence bound?

Good question…

UCB1

- Try each arm once
- At round i , select the following arm

$$\arg \max_{\{a=0,..K-1\}} \left(\hat{r}_{a,i} + \sqrt{\frac{2 \log \boxed{N}}{\boxed{n_{a,i}}}} \right)$$

Total budget

How many times arm *a* is played
up to iteration i

Find detailed analysis and theoretical proof about UCB,
and many variants of UCB1 in this paper:

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." *Machine learning* 47, no. 2 (2002): 235-256.

Applications of Multi-armed Bandit (except gambling)

- Online recommendation tasks

Player:
the recommendation algorithm

Action: make a recommendation

Environment:
the users

Reward: user feedback, e.g., click or like

Arms: candidate items to recommend

 …

The number of arms can be very large in this application. Are the algorithms mentioned still good? If not, any idea how to further improve them?

Find it interesting?

Find the answer for the last question in the following paper:

- Li, Lihong, Wei Chu, John Langford, and Robert E. Schapire. "**A contextual-bandit approach to personalized news article recommendation.**" In Proceedings of the 19th international conference on World wide web, pp. 661-670. 2010.

Books about multi-armed bandit [e-copies of both are made available by the authors]:

- Slivkins, Aleksandrs. "**Introduction to multi-armed bandits**." *arXiv preprint arXiv:1904.07272* (2019).
- Lattimore, Tor, and Csaba Szepesvári. **Bandit algorithms**. Cambridge University Press, 2020.