A/B Testing - Upper confidence bond (UCB)

Multi-arm bandit

- Inspired in the old slot machines with single lever – also known as one-armed bandits
- Strategy to explore multiple one-armed bandits and and exploit the one with the best payout
- Possible outcomes
  - Best: knowing the best payout machine and play only on this machine
  - Second best (ideal): play one time each machine and being able to define the machine with the highest payout and then play only on that machine
  - Optimized result: results of the slot machine are defined by chance. Goal - optimize the definition of the best machine (exploration) so you can spend more resources playing only in this machine (exploitation)

Approaches to Multi-arm bandit

Random

1. Exploration
   - Select the slot machines randomly or equally
2. Exploitation
   - Test each slot machine, define the best payout and then play only on this machine

Simple

1. Epsilon-Greedy
   - Every play you have an *epsilon* chance of choosing a random machine slot, and (1 - *epsilon*) of playing in the highest paying slot machine.

Dynamic

1. UCB1
   - It maximizes a function of the current payout of each machine plus a second term that considers how many times you played each machine.
   - This will give a second chance to machines that had a low payout but were not explored a lot, maybe the bad payout could be just consequence of chance.

UCB1 method

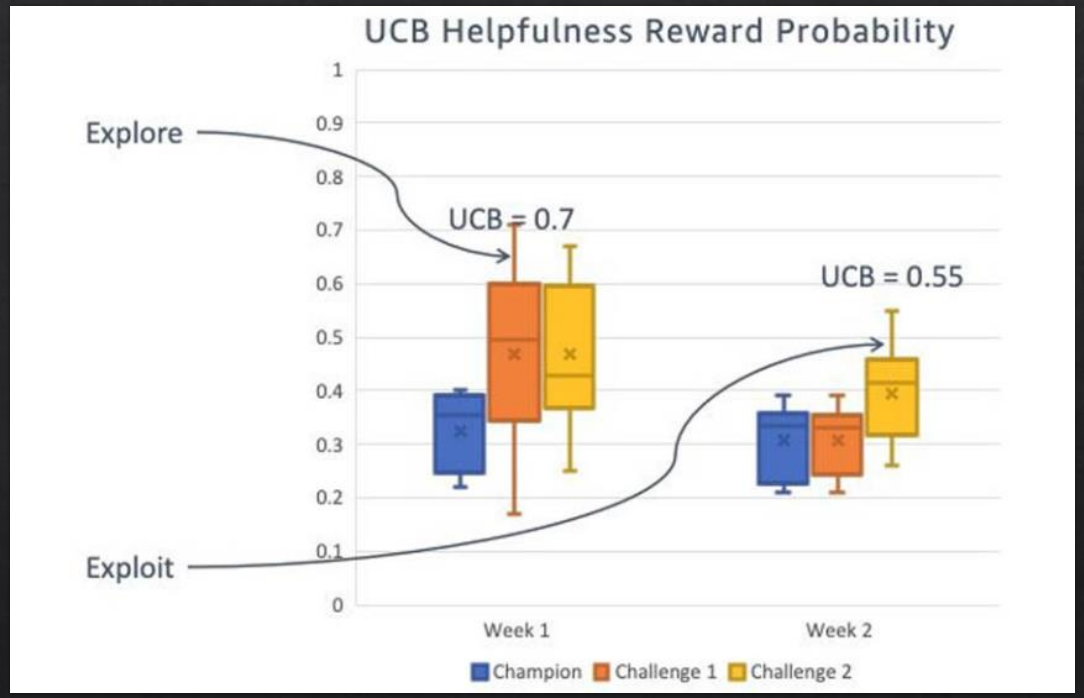
- The upper confidence bound (UCB) algorithm introduces uncertainty around variants by keeping track of how many times a variant is explored
- Consider n slot machines. Each time you play a machine i, you count how many times you played that machine n_i and record the average reward μ_i. t is the total number or rounds for all machines. The UCB1 algorithm formula to be optimized is

$$\mu_i + \sqrt{\frac{2\ln(t)}{n_i}}$$

- Second term will be greater for machines which we played less times (smaller n_i) leading to an exploration of those machines.
- UCB faces the uncertainty optimistically, it assumes that face an uncertainty the action taken is correct and it will get the highest upper bound

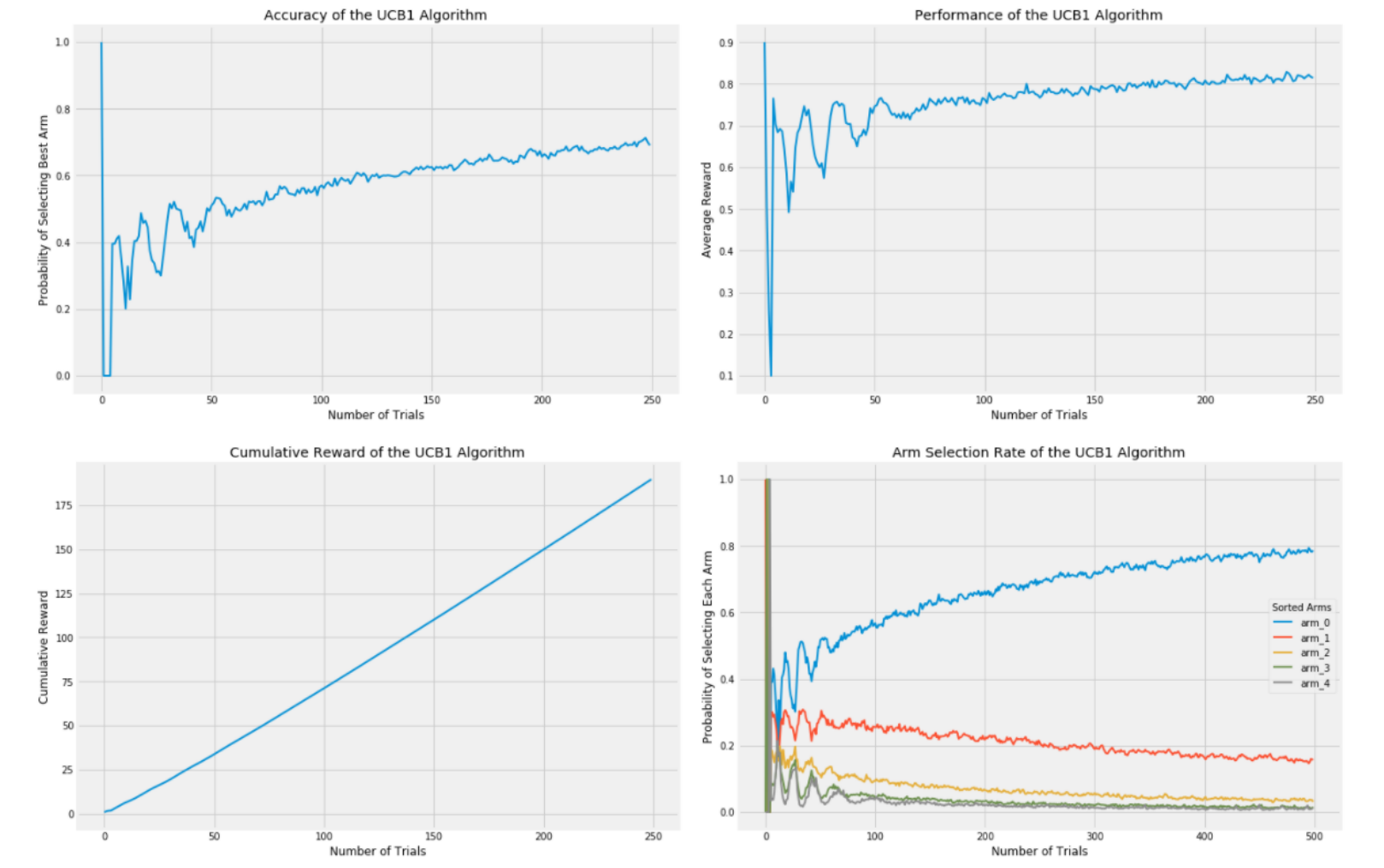
UCB reward probability

- In the following example, the Challenger 1 variant is more likely to be selected at the end of week 1 due to a higher upper confidence bound of 0.7.
- In week 2 as uncertainty levels drop, we exploit the variants with the highest mean plus uncertainty, which puts the Challenger 2 variant ahead with an upper confidence bound of 0.55



Source: https://aws.amazon.com/blogs/machine-learning/dynamic-a-b-testing-for-machine-learning-models-with-amazon-sagemaker-mlops-projects/

UCB1 performance - example



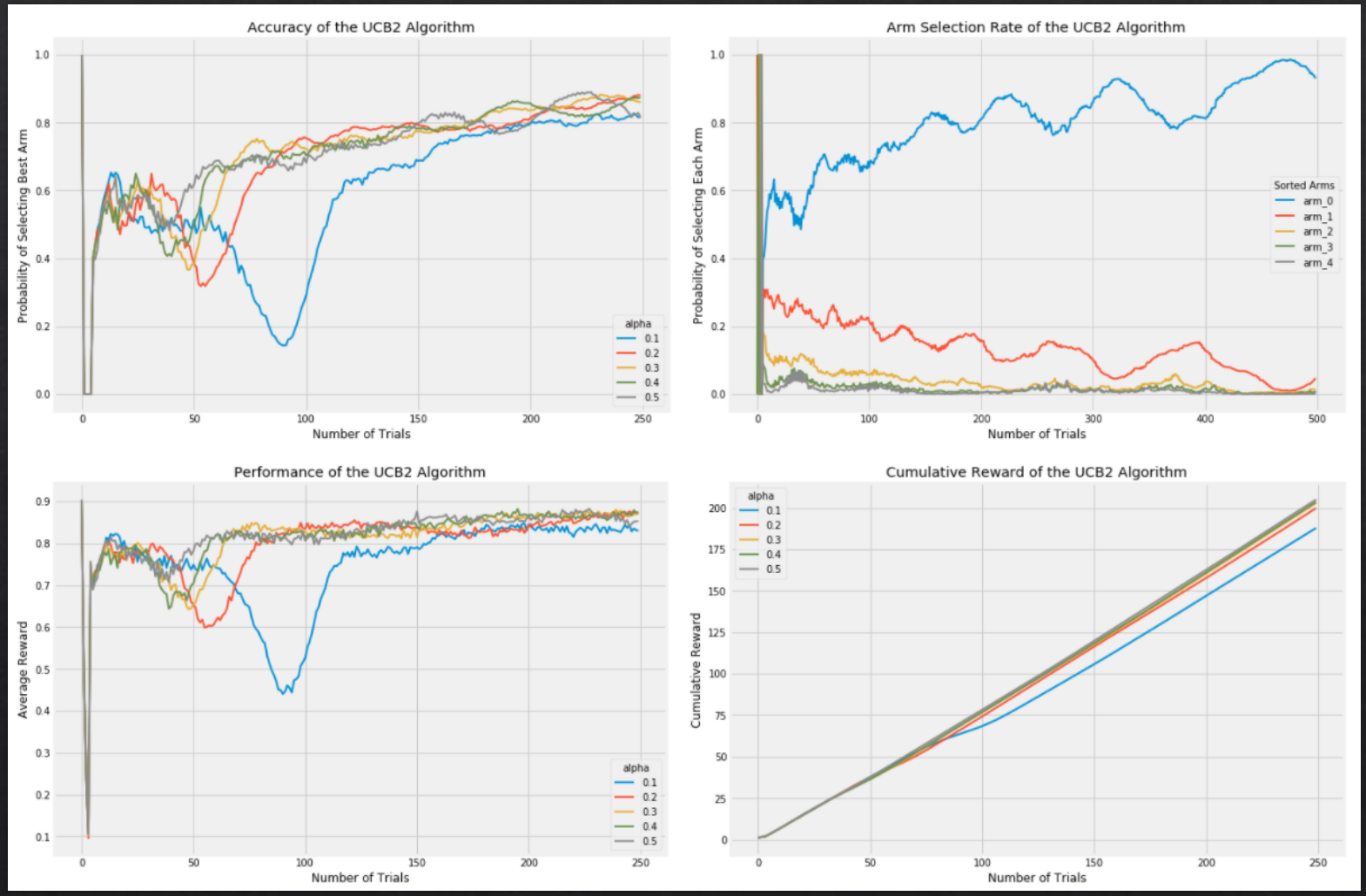
Source: https://towardsdatascience.com/a-b-testing-is-there-a-better-way-an-exploration-of-multi-armed-bandits-98ca927b357d

UCB2 algorithm

- It ensures that your best result is exploit during a certain period
- Seasonally it will explore other machines
- This behavior is especially important if the outcomes of the machines can change over time and avoid to get stuck with the initial best machine
- To achieve this a parameter α is added to tune the length while the winner machine will be exploited

Source: https://towardsdatascience.com/a-b-testing-is-there-a-better-way-an-exploration-of-multi-armed-bandits-98ca927b357d

UCB1 performance - example



Source: https://towardsdatascience.com/a-b-testing-is-there-a-better-way-an-exploration-of-multi-armed-bandits-98ca927b357d