# Report on Task 2

Leila Khaertdinova, BS2-02

May 2023

## 1 UCB1 algorithm

Upper Confidence Bound 1 (UCB1) is an algorithm that is commonly used for multi-armed bandit problems to address the exploration-exploitation trade-off. The algorithm uses an upper confidence bound (UCB) to balance exploration and exploitation while selecting which arm to play.

The UCB1 algorithm works as follows:

1. In the beginning, the algorithm plays with each arm once to gather some initial data.
2. It then calculates for each arm the mean reward and the UCB using the following formula:
$$UCB_i = \mu_i + \sqrt{\frac{2 \ln n}{n_i}},$$
where $\mu_i$ is the current estimate of the mean reward for arm $i$, $n$ is the total number of plays and $n_i$ is the number of times arm $i$ was played.
3. The algorithm selects the arm with the highest UCB value and plays with it.
4. After each arm play, the algorithm updates its estimate of the arm's mean reward.
5. Steps 2, 3, and 4 are repeated until a predetermined stopping criterion (1000 iterations) is met.

One potential limitation of the UCB1 algorithm is that it assumes that the rewards for each arm follow a stationary distribution and do not change over time. If the reward distributions shift or change, the algorithm may struggle to adapt. This is where Thompson Sampling can be advantageous.

## 2 Thompson Sampling

Thompson Sampling, unlike UCB1, can handle problems that arise when the rewards distribution changes over time. Since TS uses a Bayesian approach to estimate the reward distribution for each arm, it can dynamically update its beliefs about which arm is most likely to provide the highest reward based on new observations.

The Thompson Sampling algorithm works as follows:

1. In the beginning, the algorithm assigns a prior distribution for each arm's reward probabilities.
2. It then samples from these distributions to choose an arm to play.
3. After playing with an arm, the algorithm updates the posterior distribution for the selected arm based on the observed reward.
4. Steps 2 and 3 are repeated until a predetermined stopping criterion (1000 iterations) is met.

The key difference between TS and other bandit algorithms is in the way of balancing exploration and exploitation. UCB1 chooses the next arm to play based on its estimated potential, whereas TS chooses the next arm based on the likelihood that it has the highest reward.

## 3 Algorithms comparison

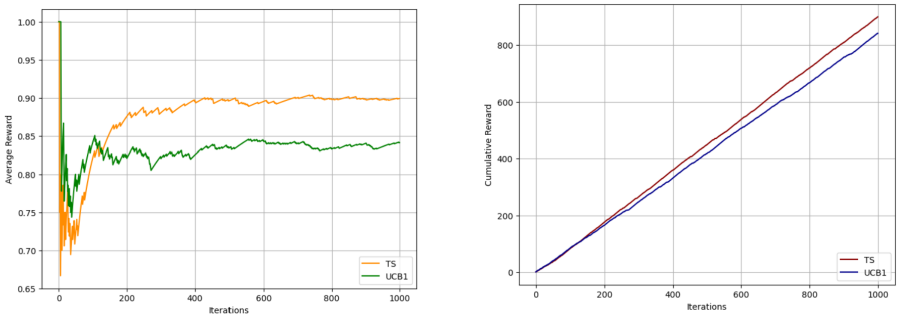
Both algorithms were considered for the following 2 cases:

1. For a three-armed Bernoulli bandit with mean rewards $\theta_1 = 0.9$, $\theta_2 = 0.8$, $\theta_3 = 0.7$ over 1000 iterations.
2. For a three-armed Gaussian bandit with mean rewards $\mu_1 = 1$, $\mu_2 = 1.5$, $\mu_3 = 1$ and stds $\sigma_1 = 2$, $\sigma_2 = 3.5$, $\sigma_3 = 1$ over 1000 iterations.

According to the obtained results that can be seen in Figure 1 and Figure 2 below, we can observe that TS provides better performance in both cases. By taking advantage of the Bayesian approach, Thompson Sampling can dynamically adapt to new reward distributions to achieve higher cumulative rewards over time as well as higher average rewards. These results suggest that Thompson Sampling appears to be more effective than the UCB1 algorithm.

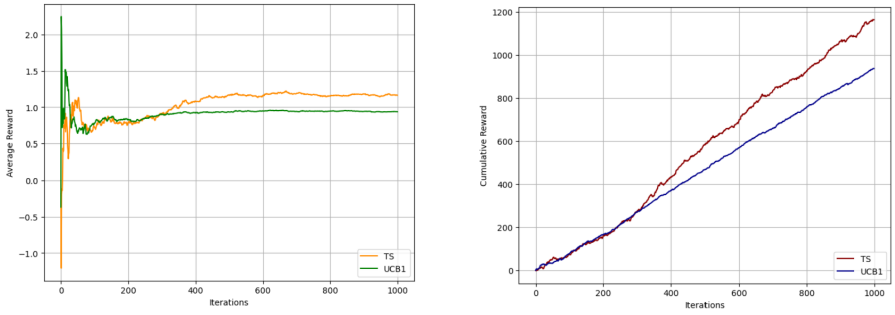
## 4 Thompson Sampling for Product Assortment

Thompson Sampling is not limited to the multi-armed bandit problem and has proven to be beneficial in solving a wide range of online decision-making problems. One such problem is product assortment planning, where TS offers an effective solution to optimizing product selection for maximum revenue.



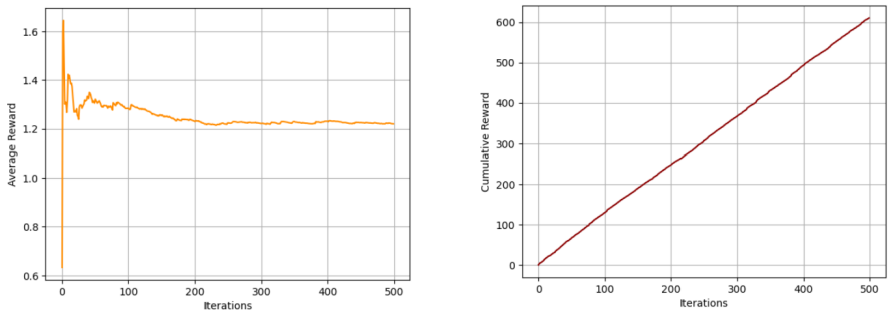

(a) Average rewards over time (b) Cumulative reward over time

Figure 1: Comparison of TS and UCB1 in case of Bernoulli rewards




(a) Average rewards over time (b) Cumulative reward over time

Figure 2: Comparison of TS and UCB1 in case of Gaussian rewards




(a) Average rewards over time

(b) Cumulative reward over time

Figure 3: Performance of TS for Product Assortment

In order to analyze the efficiency of TS in this particular problem, I tested a simulation with $n = 6$ (n - number of products), $\sigma^2 = 0.04$ (variance of demands), and profits $p_i = 1/6$ assigned to each product $i$.

The algorithm basically stayed the same, however, it was applied to a more complex problem. The prior distribution was based on the Gaussian distribution of $\theta$, with a mean of 0 with a covariance matrix where the diagonal elements had a variance of 1 while off-diagonal elements had a variance of 0.2. All other assumptions and formulas in my code implementation were taken from the TS Tutorial (attached file in Moodle).

Figure 3 presents the performance of Thompson Sampling algorithms in this problem.

In conclusion, Thompson Sampling can be applied to various complex decision-making problems besides the multi-armed bandit problem and appear to be a successful approach in Product Assortment planning.