STDSR-2023-Assignment 2 "Thompson Sampling"

Student Elina Akimchenkova
Mail e.akimchenkova@innopolis.university
Group B21-DS02

Implementation

All source code with comments and plots you can find in my github account: `https://github.com/akmchnkv/Thomson-Sampling`

Upper Confidence Bound (UCB1) algorithm

The UCB1 algorithm is a simple yet effective approach for solving the multi-armed bandit problem. It maintains an estimate of the expected reward for each arm, and uses an exploration parameter to balance between exploiting the arms with the highest estimated rewards and exploring other arms to gather more information about their rewards. The exploration parameter determines how much weight is given to exploration versus exploitation, and is typically tuned empirically.

The UCB1 algorithm provides a good balance between exploration and exploitation, and has been shown to achieve near-optimal performance in many scenarios. However, it may not always be the best choice depending on the specific problem and distribution of rewards. It's important to experiment with different algorithms and parameters to find the best solution for a particular application.

The Bandit class represents a bandit environment with a given number of arms, each having a true reward value randomly drawn from a normal distribution with mean 0 and variance 1. The pull-arm method simulates pulling a specific arm and returns a reward sampled from a normal distribution with mean equal to the true reward of the arm and variance 1.

The UCB1 class implements the UCB1 algorithm for selecting arms to pull. It maintains an estimate of the expected value of each arm and the number of times each arm has been pulled. The select-arm method calculates an Upper Confidence Bound (UCB) value for each arm, which balances the trade-off between selecting the arm with the highest expected value and exploring arms that have been pulled fewer times. The UCB value is calculated using the UCB1 formula, which incorporates the current estimate of the expected value, the number of times the arm has been pulled, and the total number of pulls across all arms. The method returns the index of the arm with the highest UCB value. The update method updates the estimate of the expected value and the count of the selected arm based on the observed reward.

The code runs the UCB1 algorithm for a bandit environment with 10 arms for a total of 1000 rounds. It keeps track of the rewards obtained and the cumulative rewards at each round, and finally plots the cumulative rewards against the number of rounds to visualize the algorithm's performance.

Thompson Sampling for Bernoulli reward

A Bernoulli bandit is a special type of multi-armed bandit problem where each arm generates a binary reward, i.e., either 0 or 1, with a probability of success p. The goal of the bandit algorithm is to find the arm with the highest probability of success.

Thompson Sampling is a probabilistic algorithm that maintains a posterior distribution over the unknown probabilities of success for each arm. In this implementation, the posterior distribution is modeled using a Beta distribution with parameters alpha and beta initialized to 1. At each round, the algorithm samples a probability of success for each arm from its posterior distribution and selects the arm with the highest sampled probability. After receiving a reward for the selected arm, the algorithm updates the posterior distribution

for that arm using Bayes' rule, i.e., by updating the alpha and beta parameters based on the observed reward.

The code initializes a BernoulliBandit object with a list of true probabilities of success for each arm. It also creates a ThompsonSampling object with the number of arms. The main loop interacts with the bandit environment for a fixed number of rounds. At each round, the algorithm selects an arm using the select-arm method of the ThompsonSampling object, pulls the arm using the pull-arm method of the BernoulliBandit object, updates the posterior distribution using the update method of the ThompsonSampling object, and records the reward and cumulative reward.

UCB and TS in case of Gaussian rewards

The UCB algorithm is based on the principle of choosing the arm with the highest upper confidence bound. The UCB algorithm keeps track of the number of times each arm has been selected, the total reward received from each arm, and the average reward received from each arm.

The Thompson Sampling algorithm is based on the principle of choosing the arm with the highest probability of being the best arm. The algorithm maintains a probability distribution over the true reward of each arm. The algorithm selects an arm by sampling from the probability distribution and selecting the arm with the highest sampled value. After an arm is selected, the algorithm updates the probability distribution over the true reward of that arm based on the observed reward.

The performance of the UCB and Thompson Sampling algorithms for the Gaussian reward case is similarly, with the Thompson Sampling algorithm slightly outperforming the UCB algorithm. The difference in performance is small and may not be statistically significant.

Correlated Travel Times

In the context of traffic routing, the multi-armed bandit problem arises when a driver is trying to find the best route to a destination in a dynamic traffic environment. In this case, each "arm" corresponds to a possible route and the goal is to find the best route to take based on the current traffic conditions. However, the travel times of different routes may be correlated due to shared congestion points, making it challenging to determine the best route using traditional methods.

Thompson Sampling provides a solution to this problem by modeling the travel times as random variables with a joint distribution that captures the correlations between routes. Specifically, the algorithm uses a Bayesian approach, where a prior distribution is placed on the joint distribution of travel times and updated based on observed data.

The key advantage of Thompson Sampling for correlated travel times is that it can handle complex dependencies between routes, making it well-suited for real-world traffic routing problems. Additionally, it is relatively easy to implement and can adapt to changing traffic conditions over time.

However, one potential limitation of Thompson Sampling is that it may require a large number of samples to accurately estimate the joint distribution of travel times, especially in high-dimensional settings. In addition, it may not perform as well as other methods when the correlations between routes are weak or poorly understood.

Overall, Thompson Sampling is a promising approach for solving the multi-armed bandit problem in the context of traffic routing with correlated travel times.