Faculty of Mathematics

Oliver Gay

Mathematics and Computer Science

The Project Title Goes Here

April 2024

Department of Mathematics

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Chapter 1

Introduction

The Multi-Armed Bandit problem (often abbreviated to MAB), is a framework in machine learning and decision theory, in which, an agent is presented with a set of actions, each with an unknown reward distribution assigned to it. The agent's objective is to attempt to maximise it's cumulative reward over a period of time by analysing the information it gathers from performing actions.

For the purpose of this paper, we define the following:

A MAB has K arms with distributions $D = d_1, \ldots, d_K$, such that arm i has distribution d_i. Let $\mu_1, \ldots, \mu_K$ be the mean of the above distributions.

For each time step $0 <= t$, an arm i is chosen, and the reward is observed as r_t. The regret ρ for time t is defined as:

$$\rho_t = \max\{\mu\} - r_t$$

And the cumulative regret γ by:

$$\gamma_t = t * \max\{\mu\} - \sum_{n=1}^{t} \rho_n,$$

1.1 The Bernoulli Bandit Problem

The Bernoulli bandit problem (where the rewards are drawn from a Bernoulli distribution of unknown mean).

1.1.1 The Gaussian Bandit Problem

The Gaussian bandit problem (where the rewards are drawn from a Gaussian distribution with unit variance and unknown mean).

1.1.2 Algorithmic Approaches

Randomized

The randomized algorithm for multi-armed bandits involves selecting arms at random with equal probability, without considering any feedback or past performance. This algorithm is commonly referred to as the "purely random" or "uniform random" algorithm.

The random algorithm can be summarized as follows:

Input: List of arms $\mathcal{A}$ with unknown probabilities $\mathcal{P}$
Output: Selection strategy

for $t = 1, 2, \ldots$ **do**
  Choose a random arm $A_t = a$ uniformly from the set of arms $1, 2, \ldots, K$;
  Receive the reward R_t by pulling arm a;
end

Algorithm 1: Randomized Algorithm

At each time step t, the randomized algorithm randomly selects an arm A_t from $\mathcal{A}$, following a uniform distribution:

$$P(A_t = a) = \frac{1}{|\mathcal{A}|} \quad \text{for all } a \in \mathcal{A}$$

This algorithm is quite often used as a baseline for more sophisticated algorithm evaluation. In the long run, the expected rewards, R_t, converge towards the average of the probabilities of each arm:

$$lim_{t\to\infty} R_t = \frac{1}{|\mathcal{A}|} \sum_{a\in\mathcal{A}} P(A_t = a)$$

Although this algorithm is very simple to implement, it doesn't attempt to optimize arm selection based on past events. It can be described as a purely exploratory algorithm that doesn't exhibit any exploitative behavior, hence its performance is generally vastly inferior to that of any other algorithms.

Greedy

The greedy algorithm is a step-up from randomly picking arms, since it actually takes into account past performance. Put simply, it chooses the arm that has the highest estimated expected reward, denoted by the arm with the highest current rate of success.

To begin, it assigns each arm with a "action-value estimate", essentially the predicted success rate of each arm. This is often set arbitrarily at some value, usually 0.5 or 1.

Then, at each time step t, the greedy algorithm selects the arm A_t from $\mathcal{A}$ with the highest expected success rate.

The greedy algorithm can be summarized as follows:

Input: List of arms $\mathcal{A}$ with unknown probabilities $\mathcal{P}$, action-value estimate $\mathcal{E}$

Output: Selection strategy

foreach *arm* $i = 1$ **to** K **do**
  Initialise success rate $Q_i \leftarrow \mathcal{E}$
  Initialise number of successes $S_i \leftarrow 1$
  Initialise count $C_i \leftarrow 0$
end

for $t = 1, 2, \ldots$ **do**
  Choose arm $A_t = a$ with the highest success rate: $a \leftarrow \arg\max_i Q_i$
  Receive the reward R_t by pulling arm a;
  Increment count: $N_a \leftarrow N_a + 1$
  Increment successes: $S_a \leftarrow S_a + R_t$
  Update success rate: $Q_a \leftarrow \frac{S_a}{N_a}$
end

Algorithm 2: Greedy Algorithm

Although this algorithm is simple and fast, it tends to exploit the immediate best option, and very rarely explores other options to try and optimise it's rewards.

It also is extremely sensitive to initial results, which significantly impacts it's performance. For example, if a 3-arm bandit has arms (0.4, 0.75, 0.8), it could get "unlucky" with reasonable probability 2% to get rewards (1, 0, 0), which effectively locks it in to keep pulling arm 0, even though the other two arms are much better.

Moreover, the value of E can significantly impact the algorithm's performance - in the extreme case where E=0, the algorithm will keep pulling the first arm that succeeds, and will only explore

other arms when the programming language registers the success rate as so low, it rounds down to 0. For example, Python will do this only when the success rate goes below $10^{-324}\%$

Epsilon Greedy

Describe high-level intuition Pseudo-code Advantages and Disadvantages.

Upper Confidence Bound (UCB)

Describe high-level intuition Pseudo-code Advantages and Disadvantages.

Thompson

Describe high-level intuition Pseudo-code Advantages and Disadvantages.

Chapter 2

Another Chapter Heading

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.