Introduction to Bandit Algorithms

Exploration-Exploitation Trade-Off of Bandit Algorithms in Comparison

submitted by

Elise Marie Wolf

born 11th April 2003

Advanced Work on the Seminar: Mathematical Methods of Artificial Intelligence

School of Business Informatics and Mathematics
Chair of Probability Theory
University of Mannheim
Supervised by Prof. Dr. Leif Döring

Mannheim
15th May 2024

Declaration of Authorship

I hereby certify that this thesis was written by me personally and that I did not use any outside help. I also affirm that this thesis or parts thereof have not been submitted by myself as proof of examination elsewhere. Verbatim or analogous copies from other writings and publications in printed or electronic form are marked. All secondary literature and other sources are acknowledged and listed in the bibliography. The same applies to graphic representations and images as well as to all internet sources. I also agree that my work may be sent and stored anonymously in electronic form for the purpose of a plagiarism check. I am aware that the work may not be corrected if this declaration is not given.

Mannheim, 15th May 2024

Place, Date

Signature

Contents

1 Introduction

Bandit algorithms are fundamental components of Reinforcement Learning (RL) systems, tasked with balancing the exploration-exploitation trade-off to optimize decision-making in uncertain environments. In this seminar work, we undertake a rigorous examination of the theoretical foundations and pragmatic implications of bandit algorithms, culminating in an in-depth analysis of the Upper Confidence Bound (UCB) algorithm.

Chapter 2 serves as a foundational exploration, beginning with an introduction to stochastic bandit problems. We begin by contextualizing the exploration-exploitation trade-off, thereby laying the groundwork for a deeper analysis of algorithmic behavior. Our investigation traces the developmental trajectory leading to the UCB algorithm, starting with the preliminary algorithms Explore-then-Commit (ETC) and the Greedy algorithm. These initial algorithms serve as stepping stones guiding our exploration towards the central focus of our inquiry: the classic UCB algorithm.

Through a systematic dissection of its components and the establishment of theoretical foundations, we provide a robust framework for understanding the UCB algorithm. Particular emphasis is placed on the derivation and validation of the regret bound as a metric for assessing algorithmic effectiveness, thereby paving the way for evaluations.

Transitioning from theoretical discussions to empirical analyses, Chapter 3 employs simulation techniques to investigate the performance of key algorithms. Here, the ETC, ε-Greedy, and UCB algorithms are implemented and evaluated using Bernoulli random variables. The resulting simulation outcomes, presented through dashboards, offer insights into the exploration-exploitation trade-off and algorithmic performance.

We highlights empirical findings in comparison with theoretical expectations, revealing the long-term superiority of the UCB algorithm in minimizing total regret. Additionally, the limitations of ETC in improving total regret become apparent, underscoring the importance of empirical validation in refining theoretical frameworks.

By integrating theoretical insights with empirical analyses, our seminar work aims to provide a comprehensive understanding of bandit algorithms, elucidating their exploration-exploitation dynamics and their practical implications in artificial intelligence research.

2 UCB Algorithm and its Basics

This chapter serves as an introductory exposition to the theoretical framework of RL. We begin by defining the conceptual framework. Subsequently, we systematically develop an understanding for the key concepts by tracing the logical development of algorithms from basic to more sophisticated forms. The chapter concludes with a detailed examination of the UCB algorithm. This is of interest in that it serves as a starting point to following chapters for a discussion on performance evaluation.

2.1 Introduction to Stochastic Bandit Problems

The aim of the first section of this chapter is intended to motivate the question of the thesis, how the choice of variance influences the exploration-exploitation trade-off. In order to address this question, we want to study the underlying model for a thorough analysis of algorithmic behaviour and outcome generation.

The comprehension of algorithms in RL relies on bandit problems, a concept introduced by William R. Thompson in 1933. These problems aim to estabilish probability criteria to discern the optimal choice of actions.[12] The first bandit problem was the **two-armed bandit machine**. In this scenario the user chooses to pull one or the other arm of the machine, which each yields a random payoff, distributed uniquely for each arm and unknown to the user beforehand. The machine got this name in memory of the one-armed bandit, an old-fashioned name describing a slot machine that relies on lever-operation.[9]

Formally defined, those problems are settings in which different decision-making options are presented as a set of arms $\mathcal{A} = \{a_1, \dots, a_K\}$, in our case finit unless stated differently. Each arm yields a certain payoff, also referred to as reward X_t. This is unknown to the user beforehand. Instead, the expectation for the return Q_a determined by a distribution P_a for each arm a can be utilized. Our objective is to maximize the expected reward over all iterations of the decision-making process by selecting the best arm more frequently. The concept of bandit problems is encapsulated within a stochastic bandit model ν and commonly reffered to as k-armed bandit, where k denotes the number of arms. The sequence of chosen distributions is referred to as a learning strategy π_t, only depending on what as been played prior to time t. [5]

Definition 2.1.1. Bandit problems in stochastic bandit models (Compare [5], Definition 1.2.1)
Suppose $\mathcal{A}$ is an index set, and $\nu = \{P_a\}_{a \in \mathcal{A}}$ is a family of real-valued distributions.

- The set ν is called a stochastic bandit model. We will always assume $\mathcal{A} = \{a_1, \dots, a_K\}$ is finite, where K is the number of arms.
- The R action value (or Q-value) of an arm is defined by its expectation $Q_a := \int x \, P_a(dx)$. A best arm, usually denoted by a^*, is an arm with the highest Q-value, i.e.,

$$Q_{a^*} = \arg\max_{a \in \mathcal{A}} Q_a.$$

Typically, one abbreviates Q^* for the largest action value Q_a, and if there are several optimal arms, the arg max chooses any of them.

- A learning strategy (or policy) for n rounds ($n = +\infty$ is allowed) consists of
  - an initial distribution π_1 on $\mathcal{A}$,
  - a sequence $(\pi_t)_{t=2,\dots,n}$ of kernels on $\Omega_{t-1} \times \mathcal{A}$, where Ω_t denotes all trajectories $(a_1, x_1, a_2, x_2, \dots, a_t, x_t) \in (\mathcal{A} \times \mathbb{R})^t$. We will write the kernels in reverse ordering of the arguments $\pi_t(\cdot; a_1, x_1, a_2, x_2, \dots, a_t, x_t)$ with the meaning that $\pi_t(\{a\}; a_1, x_1, a_2, x_2, \dots, a_t, x_t)$ is the probability arm a is chosen at time t if the past rounds resulted in actions/rewards $a_1, x_1, a_2, x_2, \dots, a_t, x_t$.

There goes one challenge in hand in defining a bandit problem: only limited information is available to the agent or learner, who chooses the next action, within the chosen environment or stochastic bandit model ν. Consequently, the environment can be viewed as an element within the set of environments $\nu \in \mathcal{E}$, from which a single environment is drawn.

To assess the efficacy of selected actions, we can compute regret, which represents the disparity between selecting the optimal actions and the actions actually chosen.

Definition 2.1.2. Regret (Compare [5], Definition 1.2.6)
Suppose ν is a bandit model and $(\pi_t)_{t=1,\dots,n}$ a learning strategy. Then the (cumulated) regret is defined by

$$R_n(\pi) := nQ^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right].$$

Definition 2.1.3. Reward Gap (Compare [5], Definition 1.2.7)
The differences $\Delta_a := Q^* - Q_a$ are called reward gaps.

The optimization problem of minimizing expected regret can be constructed as equivalent to maximizing the reward over all actions. Attaining a regret of zero necessitates that the agent must know the optimal action. However, given that only $\nu \in E$ is known, a more attainable objective is to identify a policy π such that it has sublinear regret,

$$\lim_{n\to\infty} \frac{R_n(\pi, \nu)}{n} = 0.$$

.

The regret decomposition lemma is a commonly utilized tool in regret calculation.

Lemma 2.1.4. Regret decomposition lemma (Compare [5], Lemma 1.2.8)
Defining $T_a(n) := \sum_{t=1}^{n} \mathbb{1}_{A_t=a}$, the following decomposition holds:

$$R_n(\pi) = \sum_{a\in A} \Delta_a \mathbb{E}[T_a(n)].$$

The proof involves using a clever 1 and the tower property and will not be stated here in detail. □

2.2 Development of the UCB Algorithm

In this section, we lay the groundwork by developing preliminary algorithms to comprehend the components of the UCB algorithm, which will be discussed in the subsequent chapters. The ETC algorithm serves as the fundamental approach to addressing the optimization problem, deriving intuitively. Our investigation of ETC emphasizes the basic concepts of exploration and exploitation, which are part of the later UCB algorithm. However, it becomes evident that ETC has limitations due to a lack of justification in its derivation and functionality. Subsequently, we introduce the Greedy algorithm, which prioritizes known optimal actions up to the time of execution. Nevertheless, the Greedy algorithm is primarily a stepping stone towards our focal point, the UCB algorithm, which we will delve into extensively in this thesis and further investigate in the following section.

For the evaluation of the algorithms we define the estimated action value.

Definition 2.2.1. Estimated action value (Compare [5], Definition 1.3.1)
If (A_t, X_t) is a stochastic bandit process for some learning strategy π, then we define

$$\hat{Q}_a(t) := \frac{\sum_{k=1}^{t} X_k \mathbb{1}_{A_k=a}}{T_a(t)}, \quad a \in \mathcal{A},$$

and call $\hat{Q}$ the estimated action value. $\hat{Q}_a(t)$ is the average return from playing arm a up to time t.

We employ this function to determine which arm to play in the next round. Assessing this decision through $\hat{Q}$ proves advantageous, as it allows us to estimate the true action value Q_a using $\hat{Q}_a$. By the Law of Large Numbers for independent and identically distributed (iid) samples, $\hat{Q}$ converges almost surely towards the actual value Q, making it a meaningful representation of Q. This method is referred to as the **sample-average method** [11].

We now explore potential solutions to the bandit problem by considering meaningful approaches. Prior to committing to a single option, it would be safe to test all available options to ensure the selection of the optimal one. This is precisely the objective of the basic ETC algorithm. Initially, ETC evaluates all possible actions to identify the most promising one, and subsequently allocates the remaining decisions to the arm expected to yield the highest rewards based on the initial testing phase. Consequently, our estimated action value comprises the rewards obtained by each arm during the testing phase.

The basic ETC algorithm with m rounds of testing all arms K can be written as follows. It is based on [5], Algorithm 1.

Acting greedy is how the ETC behaves for $t > mK$ and entails selecting only the arm with the highest estimated action value, which appears to offer the best reward. This approach is considered a form of Exploitation, as it prioritizes obtaining rewards while potentially sacrificing the opportunity to discover even better ones. Exploitation, therefore, leaves much of the environment unexplored. Conversely, Exploration aims to enhance estimated action values by exploring various action options and refining the reward for each through repeated selection. [10]

Algorithm 1 Explore-Then-Commit Algorithm

Data: m, n, bandit model ν, K

Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$

Set $\hat{Q}(0) \leftarrow 0$

$t \leftarrow 1$

while $t \leq n$ **do**

$$A_t \leftarrow \begin{cases} a_{t \mod (K+1)} & \text{if } t \leq mK \\ \arg\max_a \hat{Q}_a(mK) & \text{if } t > mK \end{cases}$$

Obtain reward X_t by playing arm A_t

$t \leftarrow t + 1$

The regret bound for the ETC algorithm is given through the following theorem.

Theorem 2.2.2. Regret bound for simple ETC (Compare [5], Theorem 1.3.2)
Suppose ν is a σ-subgaussian bandit model, i.e., all P_a are σ-subgaussian (see below), with K arms and $Km \leq n$ for some $n \in \mathbb{N}$, then

$$R_n(\pi) \leq \underbrace{m \sum_{a \in \mathcal{A}} \Delta_a}_{\text{exploration}} + \underbrace{(n - mK) \sum_{a \in \mathcal{A}} \Delta_a \exp\left(-\frac{m\Delta_a^2}{4\sigma^2}\right)}_{\text{exploitation}}$$

The main idea of the proof is to decompose the exploration and exploitation phase of the ETC algorithm. □

The ETC algorithm could benefit from enhancements in the reasonability of its bound. Both the parameters, n and Δ, are contingent upon the number of rounds m. When striving to minimize the regret bound through m, its selection hinges on n and Δ. Moreover, if the number of arms K exceeds the number of actions taken n, not all arm rewards can be thoroughly explored. Consequently, in such scenarios, ETC performs no better than random exploration. [5]

Another method for seeking an optimal solution, utilizing the concept of greediness, is embodied in the purely Greedy algorithm. The premise of this algorithm is to select the arm that has proven to yield the highest rewards in previous iterations. It achieves this by adding the rewards from observations until time $t - 1$ to the estimated action value $\hat{Q}(t)$, thereby estimating the best action at time t. It is evident that the initialization significantly impacts the initial value of the estimated action value $\hat{Q}(0)$ as well as subsequent estimates.

The purely Greedy algorithm is adapted from [5], Algorithm 2.

Through the definition of the purely Greedy algorithm, it becomes evident that there is a risk of focusing on suboptimal decisions to early due to insufficient exploration to discover actions with higher rewards. This behaviour is known as **committal**. [5]

Therefore, a more reasonable approach would be to incorporate random exploration alongside the purely Greedy algorithm. This additional exploration is integrated into the ε-

Algorithm 2 Purely Greedy Bandit Algorithm

Data: bandit model ν, initialization $\hat{Q}(0)$, n
Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$
while $t \leq n$ **do**
    Set $A_t = \arg\max_a \hat{Q}_a(t-1)$
    Obtain reward X_t by playing arm A_t
    Update the estimated action value function $\hat{Q}(t)$

Greedy algorithm through a random variable, which determines whether to choose a randomized or greedy action. However, this approach has its drawbacks. Firstly, if ε is chosen too small, the enforced exploration may be ineffective. Secondly, it can be proven that the exploration phase only impacts the lower bound and does not affect the upper bound of regret. Since our objective is to minimize the potential highest regret, using the ε-Greedy algorithm instead of the purely Greedy algorithm may not adequately address the main issue. We are still going to use the ε-Greedy algorithm in our further investigation to make the potential flaws comprehensible.

2.3 The classic UCB Algorithm and its Optimality

In Section 2.2, we introduced initial algorithms for bandit models aimed at optimizing reward. This section is devoted to the study of the UCB algorithm. By dissecting its components, we establish the theoretical foundation necessary to comprehend subsequent adaptations in the following chapter. The objective is to understand its functionality thoroughly, facilitating a clearer understanding of performance evaluation. Notably, we will emphasize the derivation and proof of the regret bound for this specific algorithm, as it serves as a metric to measure the effectiveness of future adaptations. It is important to note that no attempt has been made here to investigate in fine-tuning specific parameters.

In the paper of Lai and Robbins the principle commonly known as „optimism in the face of uncertainty“[9] is introduced. This principle suggests that if an arm is frequently chosen and has consistently yielded rewards with a certain confidence, one can be optimistic that selecting it again will likely result in a similar reward, even if the outcome is technically uncertain due to sampling. In practice, this entails utilizing all available data up to the latest observation to determine the next action choice. The UCB algorithm implements this principle by assigning an UCB value to each arm, which typically overestimates the mean reward and remains unknown. While it may seem counterintuitive to overestimate the average reward of arms, there is rationale behind this approach. The overestimation for a specific arm a occurs only a limited number of times because choosing arm a provides additional data that gradually adjusts the predicted average reward of arm a towards its true long-term reward, as dictated by the Law of Large Numbers for iid samples. Consequently, if a suboptimal arm a is overestimated and chosen frequently enough, its UCB will eventually converge below the UCB of the optimal arm. [8]

For the determination of the next action the algorithm uses the function

$$UCB_a(t,\delta) := \begin{cases} \infty & : T_a(t) = 0 \\ \underbrace{\hat{Q}_a(t)}_{\text{greedy}} + \underbrace{\sqrt{\frac{2\log(1/\delta)}{T_a(t)}}}_{\text{exploration bonus}} & : T_a(t) \neq 0 \end{cases},$$

which ensures that each arm is chosen at least once due to $UCB_a(0,\delta)$. Here, δ represents the error probability, intuitively serving as an upper bound for the probability that $UCB_a(t,\delta)$ underestimates the true mean. The choice of δ can be tailored to depend on n in order to derive the regret bound.

In the case of $T_a(t) \neq 0$ the $UCB_a(t,\delta)$ is split into a part that acts greedy and a bonus for choosing to explore further states, giving it the meaning of a Greedy algorithm with an extention.

To derive the choice of $UCB_a(t,\delta)$ for $T_a(t) \neq 0$, we must first establish certain properties and inequalities, all drawn from [5].

Definition 2.3.1. σ-subgaussian random variable (Compare [5], Definition 1.3.3)
A random variable X on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ is called σ-subgaussian for $\sigma > 0$, if $\mathcal{M}_X(\lambda) = \mathbb{E}[e^{\lambda(X-\mathbb{E}[X])}] \leq e^{\frac{\lambda^2\sigma^2}{2}}$ for all $\lambda \in \mathbb{R}$.

Proposition 2.3.2. (Compare [5], Proposition 1.3.4)
If X is σ-subgaussian, then

$$\mathbb{P}(X \geq a) \leq e^{-\frac{a^2}{2\sigma^2}} \quad \text{and} \quad \mathbb{P}(|X| \geq a) \leq 2e^{-\frac{a^2}{2\sigma^2}}, \quad \forall a > 0.$$

Proof. The proof is based on a trick called the Cramér-Chernoff method. The trick uses the Markov inequality for a parametrized family of functions and then optimizes over the parameter to find the best estimate:

$$\mathbb{P}(X \geq a) = \mathbb{P}\left(e^{\lambda X} \geq e^{\lambda a}\right) \leq \frac{\mathbb{E}[e^{\lambda X}]}{e^{\lambda a}} \leq e^{\frac{\lambda^2\sigma^2}{2}} e^{-\lambda a} = e^{\frac{\lambda^2\sigma^2}{2} - \lambda a}.$$

Minimizing the right-hand side for λ (by differentiation) shows that $\lambda = \frac{a}{\sigma^2}$ yields the smallest bound, and this is the first claim of the proposition. Since the same holds for $-X$, we obtain the second claim by writing

$$\mathbb{P}(|X| \geq a) = \mathbb{P}(X \geq a \text{ or } X \leq -a) \leq \mathbb{P}(X \geq a) + \mathbb{P}(-X \geq a).$$

□

Corollary 2.3.3. Hoeffding's inequality (Compare [5], Corollary 1.3.5)
Suppose $X_1, \ldots, X_n$ are iid random variables on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ with expectation $\mu = \mathbb{E}[X_1]$ such that X_1 is σ-subgaussian. Then

$$\mathbb{P}\left(\frac{1}{n}\sum_{k=1}^{n} X_k - \mu \geq a\right) \leq e^{-\frac{na^2}{2\sigma^2}} \quad \text{and} \quad \mathbb{P}\left(|\frac{1}{n}\sum_{k=1}^{n} X_k - \mu| \leq a\right) \leq 2e^{-\frac{na^2}{2\sigma^2}}, \quad \forall a > 0.$$

Proof. This follows from the proposition 2.3 because $\frac{1}{n}\sum_{k=1}^{n} X_k - \mu = \frac{1}{n}\sum_{k=1}^{n}(X_k - \mathbb{E}[X_k])$ is a centered $\frac{\sigma}{\sqrt{n}}$-subgaussian random variable. □

By applying Hoeffding's Inequality 2.3 as well as 2.3, we derive the following transformation

$$\mathbb{P}\left(\mu \geq \hat{\mu} + \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leq \delta \quad \text{for all} \quad \delta \in (0,1).$$

For a sequence of independent 1-subgaussian random variables $(X_t)_{t=1,\dots,n}$ with mean μ and $\hat{\mu} = \frac{1}{n}\sum_{t=1}^{n} X_t$. After the $t-1$ th round, the agent knows the action value of $T_i(t-1)$ samples for arm i, $\forall i = 1, \dots, k$ with the empirical mean $\hat{X}_i(t-1)$. Then utilizing the transformation mentioned before, a candidate to add to the empirical mean as an exploration bonus is $\sqrt{\frac{2\log(1/\delta)}{T_i(t-1)}}$. This choice ensures that for a lower bound δ, the chosen level of uncertainty, the mean most of the time be smaller than the estimated one. The exploration bonus of $\mathrm{UCB}_i(t-1,\delta)$ is sometimes also called confidence width. [9]
Let us now state the algorithm from [5], Algorithm 4.

Algorithm 3 UCB Algorithm

1: **Data:** bandit model ν, $\delta \in (0,1)$, n
2: **Result:** actions $A_1, \dots, A_n$ and rewards $X_1, \dots, X_n$
3: **while** $t \leq n$ **do**
4: $A_t = \mathrm{argmax}_a UCB_a(t-1,\delta)$
5: Obtain reward X_t by playing arm A_t
6: Update the estimated action value function $\hat{Q}(t)$

The reference point for assessing the algorithm's effectiveness lies in the regret bound. Consequently, focusing attention on this aspect is crucial. Hence, we not only present the theorem governing the regret bound of the UCB algorithm but also its proof.

Theorem 2.3.4. Regret bound of UCB algorithm (Compare [5], Theorem 1.3.8) Suppose ν is a bandit model with 1-subgaussian arms, $n \in \mathbb{N}$, and $\delta = \frac{1}{n^2}$. Then the UCB algorithm has the following regret upper bound:

$$R_n(\pi) \leq 3\sum_{a \in \mathcal{A}} \Delta_a + 16\log(n) \sum_{a: Q_a \neq Q^*} \frac{1}{\Delta_a}. \quad (2.3)$$

Proof. (Compare [5], Proof of Theorem 1.3.8) Before we go into the estimates let's have a quick look at the concentration inequalities again. Choosing a suitably we can make the exponential disappear:

$$\mathbb{P}\left(X \geq \mathbb{E}[X] + \sqrt{2\log(1/\delta)}\right) \leq \delta, \quad \text{and} \quad \mathbb{P}\left(|X - \mathbb{E}[X]| \geq \sqrt{2\log(1/\delta)}\right) \leq 2\delta$$

if X is σ-subgaussian. In other words, if δ is small, then a σ-subgaussian random variable takes with values in the confidence interval $\left(-\sqrt{2\sigma^2\log(1/\delta)}, \sqrt{2\sigma^2\log(1/\delta)}\right)$ with high probability. In essence, this inequality is the reason for the logarithmic regret bounds that will appear all the time, most importantly in the UCB algorithm. The concentration inequality for sums of σ-subgaussian iid random variables with mean μ (Hoeffding's Inequality 2.3) gives

$$\mathbb{P}\left(\sum_{k=1}^{n} X_k \geq \mu + \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leq \exp\left(-\frac{n}{2}\left(\sqrt{\frac{2\log(1/\delta)}{n}}\right)^2\right) = \delta$$

which will explain the choice of the exploration bonus.

We will assume again without loss of generality that $a_* = a_1$ and estimate the expected times a suboptimal arm a will be played. Combined with the regret decomposition Lemma 2.1 this will give the upper bound on the regret. For the proof we will use the random table construction of the bandit process (A, X). For that sake recall the array $\{X_t^{(a)}\}_{t\leq n, a\in\mathcal{A}}$ of independent random variables with $X_t^{(a)} \sim P_a$. One way of expressing the bandit process X_t is as $X_{T_{A_t}}^{(A_t)}(t)$. Now define $\overline{Q}_s^{(a)} = \frac{1}{s}\sum_{k\leq s} X_k^{(a)}$. We will study the event $G_a = G_1 \cap G_{a,2}$ with

$$G_1 = \left\{\omega : Q_{a_1} < \min_{t\leq n} UCB_{a_1}(t,\delta)(\omega)\right\},$$

$$G_{a,2} = \left\{\omega : \overline{Q}_{u_a}^{(a)}(\omega) + \sqrt{\frac{2\log(1/\delta)}{u_a}} < Q_{a_1}\right\},$$

with a natural number $u_a \leq n$ to be specified later. The idea of the proof lies in the following observation: If $\omega \in G_a$, then $T_a(n)(\omega) \leq u_a$.

Let's first check why (2.3) holds by assuming the contrary. Suppose $\omega \in G_a$ but $T_a(n)(\omega) > u_a$ holds. Then there must be some time index $t \leq n$ with $T_a(t-1)(\omega) = u_a$ and $A_t(\omega) = a$. Using the definitions of G_1, $G_{a,2}$, and UCB yields

$$\begin{aligned}
UCB_a(t-1,\delta)(\omega) &\overset{\text{Def}}{=} \hat{Q}_a(t-1)(\omega) + \sqrt{\frac{2\log(1/\delta)}{T_a(t-1)(\omega)}} \\
&\overset{\text{(d)}}{=} \overline{Q}(a)_{u_a}(\omega) + \sqrt{\frac{2\log(1/\delta)}{u_a}} \\
&\overset{\text{G}_{a,2}}{<} Q_{a_1} \\
&\overset{\text{G1}}{<} UCB_{a_1}(t-1,\delta)(\omega),
\end{aligned}$$

a contradiction to $a = A_t(\omega) = \operatorname{argmax}_b UCB_b(t-1,\delta)$ which is correct by the definition of the algorithm. Hence, (2.3) holds. Since $T_a(n)$ is trivially bounded by n the following key estimate holds:

$$\mathbb{E}[T_a(n)] = \mathbb{E}\left[T_a(n)\mathbf{1}_{G_a}\right] + \mathbb{E}\left[T_a(n)\mathbf{1}_{G_a^c}\right] \leq u_a + n\left(\mathbb{P}(G_1^c) + \mathbb{P}(G_{a,2}^c)\right).$$

Thus, in order to estimate the expected number of playing arms it is enough to estimate both probabilities. It now suffices to estimate separately both summands on the right-hand side of (2.3). First, it holds that

$$\begin{aligned}\mathbb{P}(G_1^c) &= \mathbb{P}\left(Q_{a_1} \geq UCB_{a_1}(t,\delta) \text{ for some } t \leq n\right)\\ &\leq \sum_{t\leq n} \mathbb{P}\left(Q_{a_1} - \sqrt{\frac{2\log(1/\delta)}{t}} \geq \hat{Q}_{a_1}(t)\right)\\ &\overset{(2.3)}{\leq} \sum_{t\leq n} \delta = n\delta.\end{aligned}$$

Next, we choose u_a which so far was left unspecified. Let us choose u_a large enough so that

$$\Delta_a - \sqrt{\frac{2\log(1/\delta)}{u_a}} \geq \frac{1}{2}\Delta_a$$

holds, for instance $u_a = \frac{2\log(1/\delta)}{\frac{1}{4}\Delta_a^2} + 1$. Then

$$\begin{aligned}\mathbb{P}(G_{a,2}^c) &= \mathbb{P}\left(\overline{Q}(a)_{u_a} + \sqrt{\frac{2\log(1/\delta)}{u_a}} \geq Q_{a_1}\right)\\ &= \mathbb{P}\left(\overline{Q}(a)_{u_a} - Q_a \geq \Delta_a - \sqrt{\frac{2\log(1/\delta)}{u_a}}\right)\\ &\overset{(??)}{\leq} \mathbb{P}\left(\overline{Q}(a)_{u_a} \geq Q_a + \frac{1}{2}\Delta_a\right)\\ &\overset{\text{Hoeffding}}{\leq} \exp\left(-\frac{u_a\Delta_a^2}{8}\right).\end{aligned}$$

Combining the above yields

$$\mathbb{E}[T_a(n)] \leq u_a + n\left(n\delta + \exp\left(-\frac{u_a\Delta_a^2}{8}\right)\right).$$

It remains to plug in. Using $\delta = \frac{1}{n^2}$ yields

$$u_a = 1 + \frac{2\log(1/\delta)}{\frac{1}{4}\Delta_a^2} = 1 + \frac{16\log(n)}{\Delta_a^2}$$

and plugging in yields

$$\mathbb{E}[T_a(n)] \leq u_a + 1 + \frac{n}{n^2} \leq u_a + 2 \leq 3 + \frac{16\log(n)}{\Delta_a^2}.$$

Combined with the regret decomposition the claim follows.

□

If the second best arm is almost as good as the best arm, the inverse reward gap influencing the bound can be quite large. Therefore the following statement of the regret bound might me more favourable.

Theorem 2.3.5. alternative upper regret bound (Compare [5], Theorem 1.3.9) Suppose ν is a bandit model with 1-subgaussian arms, $n \in \mathbb{N}$, and $\delta = \frac{1}{n^2}$. Then the UCB algorithms also has the alternative regret upper bound

$$R_n(\pi) \leq 8\sqrt{Kn\log(n)} + 3\sum_{a\in\mathcal{A}} \Delta_a.$$

Proof. The proof of the regret bound given above revealed that $\mathbb{E}[T_a(n)] \leq 3 + 16\frac{\log(n)}{\Delta_a^2}s$.

The idea of the proof is as follows. From the regret decomposition we know that small reward gaps should not pose problems as they appear multiplicatively. Separating the arms by a threshold into those with small and those with large reward gaps we only use the estimate for the ones with large reward gaps and then minimize over the threshold. Pursuing this way leads to

$$\begin{aligned}
R_n(\pi) &= \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[T_a(n)] \\
&= \sum_{a\in\mathcal{A}:\Delta_i<\Delta} \Delta_a \mathbb{E}[T_a(n)] + \sum_{a\in\mathcal{A}:\Delta_i\geq\Delta} \Delta_a \mathbb{E}[T_a(n)] \\
&\leq n\Delta + \sum_{a\in\mathcal{A}:\Delta_i\geq\Delta} \left(\frac{\Delta_a 3 + 16\log(n)}{\Delta_a}\right) \\
&\leq n\Delta + \frac{16K\log(n)}{\Delta} + 3\sum_{a\in\mathcal{A}} \Delta_a.
\end{aligned}$$

Since Δ can be chosen arbitrarily it suffices to minimize the right-hand side as a function in Δ. The minimum can be found easily by differentiation in Δ to be located at $\Delta = \sqrt{\frac{16K\log(n)}{n}}$. Plugging-in yields

$$R_n(\pi) \leq 8\sqrt{Kn\log(n)} + 3\sum_{a\in\mathcal{A}} \Delta_a.$$

□

To assess the effectiveness and performance of the UCB algorithm compared to existing ones, we can examine the regret bound of the Explore-then-Commit (ETC) algorithm. If ETC is optimally tuned, with the commitment time for each choice of Δ selected optimally, it may outperform UCB. However, if no knowledge of Δ is available and the commitment time must be determined without prior information, UCB typically performs better than ETC.

In this chapter, we began by establishing the concept behind RL and derived various algorithms. We introduced the regret bound for each algorithm to facilitate comparisons, before focusing on the UCB algorithm. This regret bound provides a foundation for further enhancements of the UCB algorithm.

3 Simulation and Empirical Analysis

Chapter 3 focuses on the practical implementation and evaluation of bandit algorithms. Specifically, we implement the ETC algorithm, the ε-Greedy algorithm as an advanced version of Greedy, and the UCB algorithm in Python. Subsequently, we provide a comprehensive evaluation of their performance using simulation techniques of Bernoulli random variables with payoffs of the arms that are close together to reveal potential weaknesses in algorithmic performance. Following this, two dashboards are available of our implemented algorithms and data on GitHub, accompaning the comprehensive evaluation.

3.1 Simulation Techniques

To evaluate the performance of the implemented bandit algorithms, we employed specific simulation techniques tailored to the nature of the problem at hand. The simulations focused on three key algorithms: ETC, ε-Greedy, and UCB algorithm. The choice of these algorithms was deliberate, aiming to encompass a range of exploration-exploitation strategies and evaluate their effectiveness under varying conditions.

In particular, the ε-Greedy algorithm was included due to its versatility and sensitivity to the choice of exploration parameter, ε. As discussed in Chapter 2, the selection of ε plays a crucial role in balancing exploration and exploitation.

Furthermore, the bandit model employed in the simulations consisted of two arms with expected rewards: The first arm with an expected reward of 0.9 and the second arm with an expected reward of 0.8. Due to simplicity of the visualization we will just consider a bandit model with two arms, but the choice of rewards is crucial. These expected rewards were deliberately chosen to be close together to create challenging conditions for the UCB algorithm. It is well-documented that UCB algorithms often struggle in scenarios where suboptimal arms are sampled frequently, as the UCB function for suboptimal arms do not significantly differ from those of the optimal arm, exacerbating the exploration-exploitation dilemma.

To simulate the arms' rewards, we utilized Bernoulli random variables, generating rewards of either 0 or 1. For the optimal arm, a reward of 1 was achieved with a probability of 0.9 and a reward of 0 with a probability of 0.1. Similarly, for the suboptimal arm, rewards of 1 and 0 were obtained with probabilities of 0.8 and 0.2, respectively. This choice of reward distribution facilitated a controlled evaluation of the algorithms' performance, allowing us to assess their ability to exploit the arms' underlying characteristics while navigating the exploration-exploitation trade-off.

3.2 Simulation of the Algorithms

To conduct a comprehensive evaluation of the implemented bandit algorithms, simulations were performed utilizing Python programming language. The simulations and their corresponding visualizations were made available in a dedicated **GitHub Repository**

(https://github.com/eelisee/introtobanditmodels).

The bandit algorithms were implemented in Python, a widely-used programming language known for its simplicity and extensive libraries for scientific computing. Python's versatility facilitated the implementation of complex algorithms and the execution of simulations with ease. Object-oriented programming principles were employed to ensure modularity and scalability across different algorithm implementations.

The simulations of the three algorithms ETC, ε-Greedy, and UCB algorithm underwent 100 rounds of evaluation to ensure robustness and consistency in the results. The bandit model employed in the simulations consisted of two arms with expected rewards, strategically chosen to challenge the UCB algorithm's ability to differentiate between optimal and suboptimal arms. Bernoulli random variables were utilized to simulate the arms' rewards, with careful consideration given to the design of the arms' expected rewards, as discussed in Section 1.

The results of the simulations were stored in separate directories for different time steps, allowing for easy access and analysis. Additionally, an `results_average` file was generated for each algorithm, providing the average values for each time step based on 100 samples. These results served as the basis for further analysis and visualization.

Interactive dashboards were developed using Plotly and Dash libraries to visualize the simulation results. The main dashboard featured three plots: total reward over time steps, total regret over time steps, and the number of optimal arm pulls over time steps, each presented on a logarithmic scale. A comparison dashboard was also provided, highlighting the frequency of 0s and 1s drawn by each algorithm at time step 100. These visualizations enabled a deeper understanding of the algorithms' performance and behavior under different conditions.

The implemented algorithms underwent rigorous cross-validation, examining metrics such as regret relative to the number of pulls of suboptimal arms and the frequency of 1s drawn compared to the total reward. This process ensured the reliability and validity of the simulation results, providing valuable insights into the algorithms' performance dynamics.

For additional details on the simulation setup, code implementation, and visualizations, please refer to the GitHub repository.

3.3 Exploration-Exploitation Trade-Off

The use of a logarithmic scale in the visualization allows for a clearer representation of the differences in performance among the algorithms. While the total reward metric appears nearly identical across the algorithms in our simulation, it's important to note that references exploring simulations with up to 1 000 000 time steps revealed a more pronounced difference. [2] However, in our simulation with a smaller time horizon, such distinctions were less apparent.

In terms of total regret, the ε-Greedy algorithm initially demonstrates superior performance, particularly up to around the 15th time step. This initial advantage can be attributed to ε-Greedy's exploration strategy, which is based on expected rewards. However, the ETC

algorithm eventually outperforms ε-Greedy. This is because ETC systematically explores all options before exploiting the best arm, resulting in a constant regret once all options have been explored. Since only two arms are available in our scenario, it quickly becomes evident which arm is the best. Therefore, ETC only needs to catch up between the 3rd and 15th time steps to outperform ε-Greedy. Conversely, ε-Greedy performs well initially due to its reliance on expected rewards, but its inclusion of exploration can lead to poorer performance once the best arm is identified, as it continues to explore unnecessarily.

Finally, the UCB algorithm starts similarly to ETC but eventually outperforms ε-Greedy after approximately 20,000 time steps. While in this specific example, ETC shows the best performance due to the chosen number of arms, it's worth noting that UCB demonstrates good long-term performance, as evidenced by the decreasing regret over time. Additionally, the proportion of optimal arm pulls exhibits a similar trend to total regret, with ε-Greedy initially performing well but diminishing over time due to random exploration, while UCB catches up and eventually surpasses ε-Greedy in terms of optimal arm pulls.

Dashboard 2 presents bar charts with two bars per algorithm, one indicating the frequency of 0s drawn and the other indicating the frequency of 1s drawn. The visualization reveals that after 100 draws, ETC performs the best, followed by ε-Greedy and then UCB. However, it's important to note that, similar to Dashboard 1 depicting total regret, the results at 100 draws don't provide a conclusive insight, as UCB eventually outperforms ε-Greedy after around 20,000 steps, as seen in the total regret analysis. Hence, it's crucial to consider longer time horizons to avoid misleading interpretations due to visual limitations, where differences become less discernible and may lead to distorted perceptions.

The conducted simulations and their results reveal interesting insights regarding the performance of various bandit algorithms. However, several unexplored aspects could yield valuable insights for optimizing and tailoring these algorithms for specific applications.

Investigating various variants of the UCB algorithm would be highly beneficial. While the classical UCB algorithm shows promising results, advanced versions such as UCB1, UCB2, and UCB-Tuned might perform better under certain conditions. UCB1, for instance, provides improved exploration by narrowing the confidence intervals, potentially leading to a quicker identification of the optimal arm. UCB2 modifies the exploration strategy by reducing the number of explorations, focusing more on arms with higher observed rewards. [3] UCB-Tuned dynamically adjusts exploration parameters based on the variance of the rewards, offering a more efficient balance between exploration and exploitation. Exploring these variants could identify specific advantages and application scenarios where a particular variant excels, thereby enhancing the robustness and flexibility of bandit algorithms in practical settings. [4]

The value of in the ε-Greedy algorithm plays a crucial role in balancing exploration and exploitation. Different ε values significantly impact the algorithm's performance. High ε values lead to increased exploration, resulting in a broader search for potential optimal arms, which is particularly useful in dynamic or unknown environments. Conversely, low ε values focus on exploitation, yielding quicker optimal results in stable environments with well-defined reward structures. Examining different ε values could provide insights

into optimal parameter settings, maximizing the algorithm's performance across various scenarios.

The number of available arms in the bandit problem directly influences the complexity of the decision-making process. A greater number of arms increases the challenge of exploration, requiring consideration of more possibilities, which might affect the performance of algorithms like UCB and ETC. Fewer options simplify the decision-making process, potentially leading to quicker convergence times and lower regret values. A systematic analysis of performance relative to the number of arms could offer valuable guidance on how algorithms should be adapted to handle different problem sizes efficiently.

The structure of expected rewards and their distributions significantly affects the performance of bandit algorithms. Closely spaced rewards make it difficult to differentiate between optimal and suboptimal arms, reducing the effectiveness of exploration. Widely varying rewards facilitate the identification of the best arms but might lead to premature exclusion of suboptimal arms. Investigating different reward structures could demonstrate how robust and adaptable the algorithms are across various reward landscapes. [4]

Specific measures to enhance the simulation include extending simulations over longer periods, such as 1 000 000 time steps, to better observe long-term trends and assess the asymptotic performance of the algorithms. Introducing more complex stochastic models that simulate realistic reward scenarios can test the algorithms' applicability in real-world settings. Additionally, utilizing parallel computations and larger datasets would increase the efficiency of simulations and obtain more robust results.

A detailed analysis and extension of simulations with the proposed improvements could yield deeper insights into the performance and adaptability of bandit algorithms. This is crucial for optimizing their application in practice, ensuring they function reliably in varying and complex environments. The outlined improvement suggestions thus provide valuable directions for future research and development in the field of bandit algorithms.

In conclusion, the detailed analysis of the Dashboards highlights the importance of considering longer time horizons for a comprehensive understanding of algorithm performance, as well as the potential implications and avenues for future research in optimizing bandit algorithms for real-world applications.

4 Conclusion

In conclusion, our seminar journey has traversed the intricate landscape of bandit algorithms, with a specific focus on the UCB algorithm. Through a rigorous examination of theoretical frameworks and empirical analyses, we have gained valuable insights into the exploration-exploitation trade-off and algorithmic performance.

We began by elucidating the foundational concepts of bandit algorithms, contextualizing the exploration-exploitation dilemma within the realm of decision-making under uncertainty. Our investigation then progressed methodically, tracing the developmental trajectory from preliminary algorithms to the classic UCB algorithm. Section 2.3 provided a detailed exploration of the UCB algorithm and its optimality, emphasizing the derivation of the regret bound as a pivotal metric for algorithmic assessment.

Transitioning from theory to practice, Chapter 3 employed simulation techniques to empirically evaluate algorithmic performance. By implementing and analyzing key algorithms such as ETC, ε-Greedy, and UCB, we gained deeper insights into their efficacy and the nuances of the exploration-exploitation trade-off.

Our empirical findings underscored the long-term superiority of the UCB algorithm in minimizing total regret, reaffirming its significance in real-world applications. Furthermore, the limitations of alternative algorithms, particularly ETC, underscored the importance of empirical validation in refining theoretical frameworks.

In synthesizing theoretical insights with empirical analyses, our seminar work has contributed to a holistic understanding of bandit algorithms and their practical implications in artificial intelligence research. Moving forward, continued research and refinement of bandit algorithms will be essential to address the evolving challenges in decision-making under uncertainty.

5 Literature

References

[1] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. 1985. URL https://www.di.ens.fr/willow/pdfscurrent/COLT09a.pdf.

[2] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009. doi: 10.1016/j.tcs.2009.01.016. URL https://www.sciencedirect.com/science/article/pii/S030439750900067X.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. URL https://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf.

[4] Olivier Cappé Aurélien Garivier. The kl-ucb algorithm for bounded stochastic bandits and beyond. 2011. doi: 10.48550/arXiv.1102.2490. URL https://arxiv.org/pdf/1102.2490.

[5] L. Döring. *The Mathematics of Reinforcement Learning*. 2023.

[6] A. Garivier and O. Cappé. Kullback-leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41:1516–1541, 2013. doi: 10.1214/13-AOS1119. URL https://arxiv.org/pdf/1210.1136.

[7] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963. doi: 10.2307/2282952. URL https://www.jstor.org/stable/2282952.

[8] T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985. doi: 10.1016/0196-8858(85)90002-8. URL https://www.sciencedirect.com/science/article/pii/0196885885900028.

[9] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 1st edition, 2020. URL https://tor-lattimore.com/downloads/book/book.pdf.

[10] A. Makone. Reinforcement learning 6: Exploration vs exploitation, 2021. URL https://ashutoshmakone.medium.com/reinforcement-learning-5-exploration-vs-exploitation-c1bae5a2ea42.

[11] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018. URL https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf.

[12] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933. doi: 10.2307/2332286. URL https://www.jstor.org/stable/2332286.