Optimistic Thompson Sampling: Strategic Exploration in Bandits and Reinforcement Learning

by

Tianyue H. Zhang

B. Sc., University of British Columbia, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

September 2023

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Optimistic Thompson Sampling: Strategic Exploration in Bandits and Reinforcement Learning

submitted by **Tianyue H. Zhang** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Mark Schmidt, Professor, Computer Science, UBC
Supervisor

Mathias Lécuyer, Professor, Computer Science, UBC
Supervisory Committee Member

Abstract

Reinforcement learning (RL) is an area of machine learning where intelligent agents take sequential actions in an environment and learn from experience to maximize rewards. This is particularly useful when the dynamics of the environment are complex and explicit training data is limited, such as in autonomous driving, robotics, game-playing, recommendation systems, finance and health care. The main challenge in RL is to balance between exploration (taking less observed actions to gather information) and exploitation (taking actions with high historical reward). Stochastic Multi-arm bandits (MABs) can be viewed as a simplified decision-making problem where the state of the agent does not change by the actions. The rich existing bandit theory provides insights into tackling expiration-exploitation trade-offs in RL.

In this work, we draw inspiration from the classical bandit algorithms upper confidence bound (UCB) and Thompson sampling (TS), and propose a novel algorithm in RL: Optimistic Thompson sampling. Optimism encourages exploration: Agents can gain information by being optimistic and playing actions that are estimated to be sub-optimal but are not sufficiently sampled. We first discuss our performance metric in theoretical analysis, namely regret. Regret measures how the algorithm performs compared to the optimal strategy if the rewards and dynamics of the environment are known. We then provide a novel theoretical analysis of the optimistic Thompson sampling (O-TS) [Chapelle and Li, 2011] algorithms in bandits. Next, we extend the algorithms to the Markov decision process (MDP) setting, a common representation for RL problems. We propose two novel algorithms, O-TS-MDP and O-TS-MDP+. We compare their performance to existing work both theoritically and with numerical experiments. Finally, we conclude our work with a discussion of future directions for strategic exploration in bandits and RL.

Lay Summary

We often need to make a series of decisions and do not have the full information. Doctors adapt medical treatment plans according to patient responses; self-driving cars navigate uncertain road conditions by continually making decisions based on sensor signals; bankers adjust financial investments over time depending on market fluctuations. Optimizing decision-making involves the trade-off between exploration (trying out unfamiliar options to gather information) and exploitation (taking the best possible actions under known information). In this work, we design ways to make a series of decisions to balance this trade-off effectively and efficiently. The key ingredients in our methods are optimism and randomization. We encourage the exploration of uncertain options through optimistic estimates of their payoffs and introduce randomness to prevent getting stuck in less ideal choices, thus fostering the discovery of better solutions over time. We demonstrate the performances of our methods with mathematical proofs and computer experiments.

Preface

This thesis is based on the collaborative work Hu et al. [2023] with Dr. Bingshan Hu (University of Alberta; Amii), Dr. Nidhi Hegde (University of Alberta; Amii), and my supervisor Dr. Mark Schmidt (University of British Columbia; Amii). It was published at the Uncertainty in Artificial Intelligence (UAI) conference in 2023 with the title *Optimistic Thompson Sampling-based algorithms for episodic reinforcement learning*.

The breakdown of the contribution for the paper is as follows: There are four algorithms included in the paper: O-TS and O-TS$^+$ for bandits, and O-TS-MDP and O-TS-MDP$^+$ for RL. The idea of O-TS in bandits is first empirically evaluated by Chapelle and Li [2011]. Dr. Hu proposed the original ideas for the other three algorithms and developed the theoretical analysis for all four algorithms. I contributed to the implementation of the numerical experiments, comparison to existing work, and the related writing and visualizations. I attempted to analyze the two bandit algorithms, but due to limited time, the analysis was completed by Dr. Hu in the paper. Dr. Scmidt and Dr. Hegde have provided valuable edits and comments to improve the clarity and presentation of the work.

The goal of the thesis is to define and motivate the research problem, and to describe the dense theory of the paper in a way that is intuitive and approachable to researchers who are unfamiliar with this line of work. The writing and visualizations in this thesis, unless specified otherwise, are the original product of the author Tianyue H. Zhang. Dr. Schmidt, Dr. Hu, and Dr. Mathias Lécuyer have provided valuable edits throughout the process.

In Chapter 2, I include a summary of different definitions of regret. This is the part I found most confusing when I was first introduced to the topic, so I hope it could be a useful clarification. In Chapter 3, I include the proof intuition for two well-known bandit algorithms, UCB Auer et al. [2002a] and Thompson sampling Agrawal and Goyal [2017], and discuss my own understanding of our analysis for O-TS and O-TS$^+$. In Chapter 4, I draw connections between bandits and RL and discuss other types of RL theory work and their assumptions. I find it helpful to understand different perspectives for approaching the same problem. In Chapter 5, I discuss our proposed RL algorithms and their empirical performance. The writing in this chapter is partially borrowed from the paper. Finally, in Chapter 6 I conclude the thesis with a few future directions. The fast algorithm with greedy updates in this Chapter is proposed by Dr. Hu and I performed empirical evaluations. The connection between differential privacy is proposed by Dr. Lécuyer and is an ongoing project with all collaborators mentioned above.

Contents

List of Tables

List of Figures

Acknowledgments

I am grateful to my supervisor, Dr. Mark Schmidt, for admitting me as his student, for encouraging me to think independently, and for giving me the freedom to pursue my interests. Dr. Schmidt has always been patient and supportive and believed in my abilities even when I was in doubt myself. I am lucky to work with Dr. Bingshan Hu, who patiently guided me through this work and taught me the process of completing a research project. In addition, I would like to thank many other professors that I have collaborated or interacted with throughout the degree, including Dr. Sharan Vaswani, Dr. Michiel van de Panne, Dr. Mathias Lécuyer, Dr. Danica J. Sutherland, and Dr. Nidhi Hegde. They all have provided me with guidance and encouragement throughout this journey. The thought of being someone like them one day and passing on the inspiration motivates me to continue on the path of academic research.

I am blessed to be part of the family of ml-in-568. I am very thankful to all my colleagues and friends including Alan, Fred, Greg, Betty, Dylan, Curtis, Wilder, Nick, Victor, Sophie, and many many others. All of them have provided me with support and company both academically and personally and made my time here enjoyable and memorable. I am also thankful to all the previous members, including Aaron, Cathy, and Reza, for their guidance and example.

Outside of academia, I am very lucky to have many friends who supported me and have grown up with me. Thank you, Annie, Berk, and Ray, for the substantial support as well as the constructive criticism when I needed it. I am also lucky to have my cat, Cashew, for the consistent company.

I am deeply grateful to my parents, Xuxiang Zhang and Yanhong Li, for their unconditional love and faith in me. They have worked hard to provide me with the best support and resources that they could. They have set excellent examples through their actions yet still gave me the freedom to choose my own path. Due to current circumstances, they have been on the other side of the earth throughout my entire degree. I hope to see them soon.

Finally, I would like to thank all the researchers whose work I have cited in this thesis. I'm grateful to be able to contribute to this line of work. These two years of my Masters have been a challenging time, with the chaos caused by the global crisis and national events, as well as unexpected changes in my own career and personal life. In a sense, part of the philosophical idea behind this project, *Optimism in the Face of Uncertainty*, has helped me to navigate through these challenges. I hope to always keep an open mind for new experiences and move forward to the next chapters in life with an optimistic outlook.

Chapter 1

Introduction

Reinforcement learning (RL) is a subfield of machine learning that focuses on sequential decision-making under uncertainty. RL gained its modern interest in 2016 when DeepMind developed a groundbreaking AI gaming program named AlphaGo [Silver et al., 2016]. It gained significant attention when it defeated the world champion Go player, marking a major milestone in AI and game playing. The agent was trained and improved using RL and played many games against itself to refine its strategies. Since then, the field of RL has continued to expand with numerous and diverse practical applications. In robotics, agents are trained with RL to complete tasks like grasping objects [Gu et al., 2017] or navigating through space [Zhu and Zhang, 2021]. In the financial sector, RL is used for portfolio optimization [Soleymani and Paquet, 2020] and algorithmic trading [Li et al., 2019]. In self-driving cars, RL is employed to take action in real-time while navigating through traffic safely [Sallab et al., 2017, Shalev-Shwartz et al., 2016]. In healthcare, researchers use RL to design personalized treatment recommendations [Coronato et al., 2020], discover new drugs [Popova et al., 2018], and optimize resource allocation [Yu et al., 2021]. In the recent development of large language models such as GPTx [Ouyang et al., 2022], RL is also used to collect human feedback to design better dialogue responses.

Inspired by behavioural psychology, the RL agent interacts with the environment and receives feedback through rewards or penalties based on the actions it takes. The goal is to learn a policy that determines the sequence of actions to take in different states of the environment to maximize its cumulative rewards over time. Because of its adaptability to environmental changes and its ability to learn autonomously from experience without manual programming, RL is well-suited for handling complex, dynamic, uncertain decision-making tasks in real life that are difficult to model explicitly.

Despite its broad application, RL algorithms face many challenges and are far from perfect. Agents often require many interactions with the environment to learn effective policies, which can be expensive and time-consuming in real-world scenarios. In addition, balancing exploration and exploitation is a fundamental challenge: the agent needs to explore different actions to learn about their effects, but it also needs to exploit its current knowledge to maximize rewards. Moreover, it isn't easy to guarantee that RL algorithms converge to optimal or near-optimal policies, especially in complex environments. Therefore, it is essential to design

reliable and efficient algorithms that leverage historical data to collect new data that optimizes reward and information gain.

Regret analysis in RL theory helps address these challenges by providing a framework to quantify the performance. Although many theoretical studies, including our work, make simplifying assumptions about the environment and may not fully capture the complexities of real-world scenarios, they often provide valuable insights. Markov Decision Processes (MDPs) are a simplified representation of RL that are commonly used in the literature. The simplification allows researchers to analyze the convergence and optimality of algorithms under certain conditions and quantitatively evaluate and compare algorithms. Establishing the performance and reliability of algorithms allows researchers and practitioners to make informed decisions in algorithm designing and use available data more efficiently.

Stochastic Multi-arm bandits (MABs) can be viewed as a simplified subset of MDPs with no states. Inspired by slot machines in gambling, an agent in a bandit environment faces a series of actions, each associated with an unknown reward, and can only learn the reward of the action it selects. The objective is to make decisions under uncertainty while minimizing regret (the difference between the reward received and the reward that could have been received with perfect knowledge). Unlike in MDPs, the actions taken in the bandit problems do not impact its state in the environment. The advantage of considering bandit problems is that they have been studied extensively for many decades, dating back to the mid-20th century [Robbins, 1952]. The rich existing bandit literature provides important understandings in tackling expiration-exploitation trade-offs and can be a powerful source of inspiration for algorithms in the more general MDP settings.

Besides providing insights into MDPs, studying bandit algorithms on their own is already valuable in practice where data and resources are limited, and efficient decision-making is essential. Bandit algorithms could be applied in adaptive clinical trials to determine the most effective treatment among multiple options while minimizing the number of patients exposed to ineffective treatments. In online advertising, bandit algorithms are used to decide which ad to display to users based on limited information about the effectiveness of each ad. Bandits are also used in personalized recommendation systems to determine which items to show users to maximize engagement.

In this work, we revisit the classical bandit algorithms: upper confidence bound (UCB) and Thompson sampling (TS). We also provide a novel theoretical analysis of *optimistic* Thompson sampling (O-TS) algorithms that are based on a combination of TS and UCB. Optimism encourages exploration: Agents can gain useful information by being optimistic and playing actions that are estimated to be sub-optimal but are not sufficiently sampled. We then extend the algorithms to the MDP setting and propose two novel algorithms, O-TS-MDP and O-TS-MDP$^+$. We compare their performance both theoritically and with numerical experiments. Finally, we conclude with a discussion of existing work and provide directions for future studies of strategic exploration in bandits and RL.

1.1 Organization and Contribution

The organization of this thesis and the contributions in each chapter are as follows:

In Chapter 2, we formally define the stochastic multi-armed bandit problem and its performance measurement, namely the regret. There are many different definitions of regret studied in the related literature. Some are due to practicality, for example, pseudo-regret is easier to analyze than actual regret; some are due to different assumptions and perspectives of the problem, for example, the worst-case regret versus the problem-dependent regret. We will explain the difference and connections between these regret definitions, and justify our choice of performance measurement. We will discuss what it means for a policy to achieve optimality or near optimality under the measurement that we choose.

In Chapter 3, we will discuss the exploration-exploitation trade-off in bandits, and why it is challenging but essential to design algorithms with good performance. We introduce some of the existing approaches to tackle this trade-off. One of the most widely used exploration schemes, epsilon-greedy, is not optimal both theoretically and in practice. The UCB [Auer et al., 2002a] algorithm is based on the philosophy of optimism in the face of uncertainty. It is a simple algorithm to implement and enjoys elegant theoretical analysis on its regret upper bound. Thompson's sampling [Agrawal and Goyal, 2017] is another provably efficient algorithm that approximates each arm's reward by iteratively constructing posterior distributions based on the collected data. We also include a brief proof sketch of these two algorithms to serve as the foundation for our proposed algorithm and its analysis. Finally, we introduce optimistic Thompson sampling (O-TS) and O-TS$^+$. The O-TS algorithm is a modified version of the TS algorithm that is more optimistic. It was first empirically evaluated by Chapelle and Li [2011], and here we fill in the gap in theoretical analysis. Here, we relate the proof of O-TS to Thompson's sampling analysis in both worst-case and problem-dependent settings. O-TS$^+$ on the other hand is a more optimistic clipping of TS that can be viewed as a randomized version of UCB, so we discuss how the analysis is developed from the UCB proof. Finally, we demonstrate these methods' empirical performance by running a numerical experiment on a simple bandit example.

In Chapter 4, we discuss reinforcement learning and its commonly used framework, Markov Decision Processes (MDPs). We provide formal definitions and notations and discuss the relationship between bandit and MDP. We explain some common assumptions in the literature, such as stationary versus time-dependent, finite horizon versus infinite, stochastic policy versus deterministic, etc. The discussion is to clarify the assumptions we made in this work, where we mainly focus on the finite horizon, time-inhomogeneous, episodic MDPs. Exploration algorithms in RL often consist of two phases: planning the optimal policy based on existing data, and sampling based on the planned policy to collect new data. As the planning phase is trivial in bandit, we discuss more in detail some fundamental algorithms for planning when the underlying MDP reward and transition dynamics are known. We discuss dynamic programming in the finite horizon, value iteration, policy iteration, and linear programming in the infinite discounted horizon. Then, we include a literature review of the exploration algorithms in RL, mainly how the ideas from bandit algorithms such as UCB and TS have been adapted and extended to the MDP setting. We discuss how the two classes of algorithms relate or compare to each other before introducing our algorithms in the next chapter.

In Chapter 5, we introduce two novel algorithms for finite horizon, episodic MDPs, O-TS-MDP and O-TS-MDP$^+$. These algorithms are a natural extension of our bandit algorithms in Chapter 3. They rely on a modified version of TS-like posterior distribution generated from historical data and carefully chosen variance that reflects uncertainty. We describe the algorithm in detail and discuss the intuition behind the improved performance and regret bounds. Then, we run numerical simulation experiments to demonstrate their performance. We compare the original, unclipped version of Thomspon sampling and our two clipped versions. We also compare them with other existing work including SSR-Bernstein [Xiong et al., 2022] and UCB-VI Azar et al. [2017].

In Chapter 6, we summarize and conclude our work and contributions, advantages, and limitations. In the future work section, we include a few ideas related to and worth exploring. Firstly, in this work, we focus on designing good reward distributions. The transition dynamics in MDPs are another source of uncertainty so it is worth exploring how to take advantage of that. Secondly, it is of interest to practitioners how the theory can be applied to practice, so we include a few modifications of these algorithms and practical approaches in deep learning. Then, we also explore existing work in exploration in distributional RL, which can be generalized into deep learning more easily, and explore a greedy update rule to improve computational complexity. Finally, we discuss the possible connections between exploration in RL and differential privacy. We believe this novel connection can potentially yield valuable theoretical advancement and practical impact in both fields.

Chapter 2

Stochastic Multi-Arm Bandits

A multi-arm bandit (MAB) problem is a classic decision-making problem in which a player must choose between several options (or arms), in a sequential manner, with the goal of maximizing their cumulative reward over time. The term "bandit" comes from the idea of a gambler pulling the arm of a slot machine (or "one-armed bandit") in order to try and win money. While the history of the multi-arm bandit problem dates back to the 1950s when it was first introduced as a mathematical model [Robbins, 1952], the problem has since been studied extensively in the fields of probability theory, statistics, optimization, and machine learning [Lattimore and Szepesvri, 2020].

Stochastic multi-arm bandits have a wide range of applications in various fields including clinical trials [Villar et al., 2015], financial investments [Bouneffouf and Rish, 2019], and perhaps most famously online advertising [Schwartz et al., 2017]. For example, a company has a few different advertisements that can be displayed to the target audience, and there is a monetary reward each time the user clicks on the ad. The likelihood of a user clicking on each ad is unknown, but we can model this problem using a multi-arm bandit and design algorithm that monitors the performance of each ad and optimizes advertising spend by showing the most effective ads more frequently.

In the stochastic multi-arm bandit problem, each arm has an associated probability distribution of rewards, and the player must choose which arm to pull at each time step without knowing the exact distribution of each arm. The goal is to find the arm with the highest expected reward while balancing the trade-off between exploration and exploitation. Exploitation involves pulling the arm that is known to produce high rewards based on past experience, whereas exploration involves pulling arms that may produce lower rewards but have not been extensively explored yet. Balancing between the two is crucial to maximizing long-term rewards: if the agent only focuses on exploitation, it may miss out on discovering better options; if it only focuses on exploration, it may waste time on the sub-optimal arms.

Regret analysis is often used to provide a quantitative measure of algorithm performance in multi-armed bandits. Generally speaking, regret is a measure of how much the agent's chosen actions deviate from the optimal actions. It represents the opportunity cost of not always choosing the best action. With this tool in mind, researchers can design algorithms and strategies that balance the exploration-exploitation trade-off.

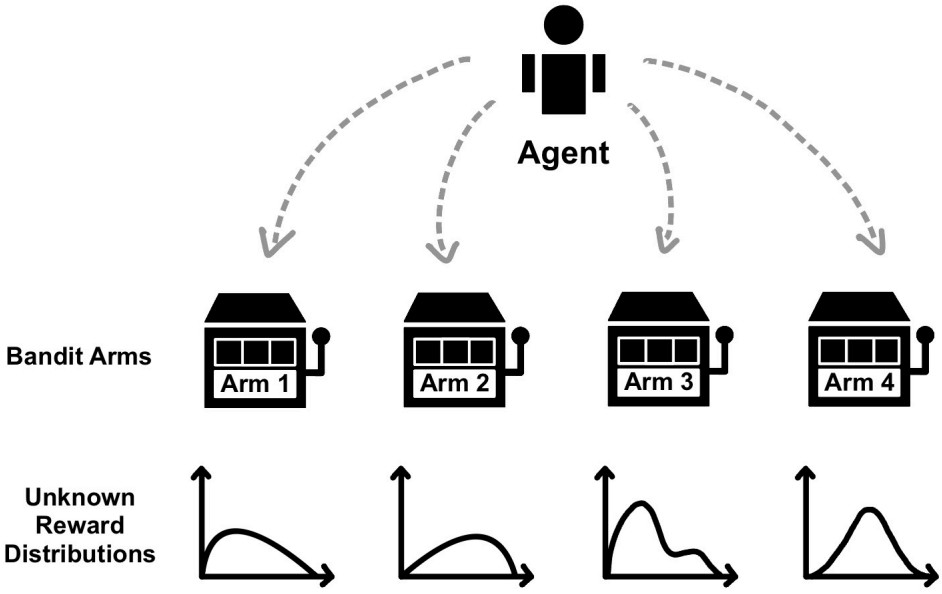

Figure 2.1: Multi-armed Bandits

A good algorithm should minimize regret or equivalently maximize the reward.

In this chapter, we will formally define the multi-armed bandit (MAB) problem. Then, we will introduce a few different forms of regret, and what it means for an algorithm to be optimal in each metric. We will discuss the advantages and drawbacks of each performance measure and justify our choice in the rest of this work.

2.1 Problem Definition

Consider the standard bandit problem where we have a set of K arms. Each arm $j \in [K]$ has an unknown reward distribution with bounded support. Without the loss of generality, we assume that the reward is in $[0,1]$. Let μ_j be the mean reward of the arm j, in other words $\mu_j = \mathbb{E}[X_{j_t}]$. We also assume there is no dependency between arms. At each round t, the agent plays an arm $J_t \in [K]$ and receives a reward $X_{J_t}(t)$. Let $\mathcal{F}_t = \{J_\tau, X_\tau(\tau), \tau = 1, \ldots, t\}$ be the history. The agent interacts with the bandit according to a sampling strategy that is learned based on the history, which we call a policy $\pi \in \Pi$. See Table 2.1.

Table 2.1: Notations for Multi-arm Bandits

K	Number of arms
μ_j	Mean reward of arm j
J_t	The arm played in round t
Δ_j	Sub-optimality of arm j
$X_{J_t}(t)$	Reward received by playing J_t
$O_j(t)$	Number of times arm j has been played by round t
$\mathcal{F}_t$	History by the end of round t
$\pi \in \Pi$	Policy that determines the sequence of actions

We consider a finite horizon of T rounds. The goal of the agent is to maximize the cumulative reward

$$\sum_{t=0}^{T} X_{J_t}(t)$$

Or equivalently, minimize cumulative regret, which is the difference between the reward from the played arm at each round, and the best possible reward at each round. Let J_t be the arm played in round t. The *regret* of an algorithm π after playing T rounds is defined as

$$Regret_T(\pi) = \max_{i=1,\ldots,K} \sum_{t=1}^{T} X_{i_t} - \sum_{t=1}^{T} X_{J_t}.$$

It is useful to consider the regret of an algorithm since it quantifies the performance of an algorithm compared to the best outcome if the reward model is known.

We often analyze the performance of an algorithm with the expectation of regret over the randomness in rewards, where lower regret indicates superior performance. The *expected regret* is defined as

$$\mathbb{E}[Regret_T(\pi)] = \mathbb{E}\left[\max_{i=1,\ldots,K} \sum_{t=1}^{T} X_{i_t} - \sum_{t=1}^{T} X_{J_t}\right]$$

The agent aims to minimize regret by learning a policy of which arm to play next based on past observations. However, the expected regret can be difficult to analyze and optimize. In this work, we instead try to minimize the *pseudo-regret*, a weaker notion that is commonly used in many other related works. The pseudo-regret of a policy is defined as

$$R_T(\pi) = \max_{i=1,\ldots,K} \mathbb{E}\left[\sum_{t=1}^{T} X_{i_t} - \sum_{t=1}^{T} X_{J_t}\right]$$

Notice that $\mathbb{E}[Regret_T(\pi)] \geq R_T(\pi)$. Without loss of generality, we can assume the first arm is the unique optimal arm, meaning $\mu_1 > \mu_j \forall j \neq 1$. Using that $\mathbb{E}[X_{J_t}] = \mu_{J_t}$, we can write the pseudo-regret as:

$$R_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} X_{1_t}\right] - \mathbb{E}\left[\sum_{t=1}^{T} X_{J_t}\right] = T\mu_1 - \mathbb{E}\left[\sum_{t=1}^{T} X_{J_t}\right]$$

Let $\Delta_j = \mu_1 - \mu_j$ to be the sub-optimality gap (or instantaneous regret) of arm j. Then, we can further decompose the regret:

$$\begin{aligned} R_T(\pi) &= T\mu_1 - \mathbb{E}\left[\sum_{t=1}^{T} X_{J_t}\right] \\ &= \sum_{j=1}^{k} (\mu_1 - \mu_j)\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[J_t = j\right]\right] \\ &= \sum_{j:\Delta_j > 0} \Delta_j \mathbb{E}\left[O_j(T)\right] \end{aligned}$$

where $O_j(t) = \sum_{t=1}^{T} \mathbb{1}\left[J_t = j\right]$ is the number of times arm j has been played by time t. This decomposition will be useful when we consider problem-dependent regret bounds, which will be discussed more in detail in the next section.

2.2 Performance Measures

There will always be some regret given the agent is unaware of the reward distributions, and it is important to talk about the lower bound of the regret. This can answer the question of how good one can do in theory. From there, we can then discuss the notion of optimality under different assumptions. For example, do we want an algorithm to be not too bad for any bandit problems, or do we care about being better for some easier bandit problems at the expense of being bad for harder problems? Since the rewards are stochastic, do we care about the performance in the worst cast, or on average? Do we care to distinguish whether the algorithm makes a few large mistakes or many small mistakes? In this section, we will discuss a few types of regret measures and how they approach these questions.

Worst-case (problem-independent) regret and optimality In the previous definitions for simplicity, we omitted the dependence on a specific bandit environment Θ. The worst-case regret of a policy π over T rounds is a problem-independent performance measurement, meaning it considers the hardest problem" in a set of bandit environment Θ:

$$\sup_{\theta \in \Theta} R_T(\pi, \theta)$$

The terms problem-independent regret and worst-case regret are interchangeable. Both capture the regret that an algorithm can incur over all possible bandit instances of a given number of arms. These measures do not consider any particular structure or correlation between the arms.

Minimax regret and optimality While worst-case regret measures the performance of a single policy in a given bandit environment, minimax regret measures the performance of a set of policies. Let Π be the set of all policies, then the minimax regret is

$$R_T^* = \inf_{\pi \in \Pi} \sup_{\theta \in \Theta} R_T(\pi, \theta)$$

The word "mini-max" comes from minimizing regret over the set of policies, while maximizing regret over the bandit environment. Minimax regret is not just a property of a policy, but also a property of the bandit problem itself: a large minimax regret could mean that the bandit environment is difficult to learn in the worst case. A policy is called minimax optimal when $R_T^* = R_T(\pi)$.

In the case where $T \geq K$, Auer et al. [1995] shows that all policies suffer $\Omega\left(\sqrt{KT}\right)$ worst-case regret, which is called the *minimax-optimal* regret. Achieving minimax optimal is typically difficult in practice, and we often settle for near-optimal algorithms.

Problem-dependent regret In practice, mini-max optimal policies can be too conservative as they do not take advantage of the specific structures or difficulties of a bandit instance. It is useful to look for policies that perform well in easier problems. Problem-dependent regret, denoted as $R_T(\pi, \theta)$ refers to the regret that a specific MAB algorithm incurs on a particular set of arms or distributions. This type of regret takes into account the specific characteristics of the problem, such as each arm's sub-optimality Δ_j. In general, problem-dependent regret is usually lower than problem-independent regret because an algorithm can exploit the specific characteristics of the problem to achieve better performance.

Asymptotic problem-dependent optimality An algorithm is called strongly consistent if they have sub-polynomial regret on all problems. Existing literature [Burnetas and Katehakis, 1997, Lai and Robbins, 1985] showed that there is a regret lower bound and that any consistent policies cannot do better. Consequently, a policy π is *asymptotically optimal* if

$$\lim_{n \to \infty} \frac{R_T(\pi)}{\log(T)} = \sum_{j:\Delta_j > 0} \frac{2}{\Delta_j}$$

Lai and Robbins [1985] also showed that it is possible to construct an algorithm that achieves asymptotic optimality. It's worth noting that asymptotically optimal policies can still have a long burn-in period or large mini-max regret.

Finite-time problem-dependent optimality Finite-time lower bounds can be established by making a finite-time analogue of consistency. Without going into the details, **?** showed that any algorithm that has reasonable worst-case performance cannot have much better problem-dependent regret than UCB.

Bayesian Approach All the above-mentioned concepts (and our work in the following chapters) are based on the stochastic approach, where we don't make assumptions about the reward distribution. In comparison, the Bayesian approach assumes that the reward of each arm is drawn from a family of parameterized distributions. Here, Bayesian regret is used to measure the expected cumulative regret, taking an expectation over the prior. The Gittins index theory proposed by Gittins and Jones [1979] provides a method of prioritizing arms that can maximize the discounted cumulative reward. The prior assumption and discount factor make things easier and allow convergence analysis, but choosing the proper assumptions can also be challenging in practice. In this sense, the stochastic approach can be more robust.

Adversarial Approach The adversarial approach can be thought of as a two-player game played by a forecaster and an adversary. At each round, the forecaster tries to play an action to minimize regret, but the adversary gets to determine the reward of each arm in order to maximize regret. In other words, the cumulative regret in this approach is

$$Regret_T(\pi) = \max_{i=1,\dots,K} \sum_{t=1}^{T} X_{i_t} - \sum_{t=1}^{T} X_{J_t}.$$

This is a harder problem to optimize compared to the stochastic approach. Adversarial algorithms such as Exp3 [Auer et al., 2002b] are also robust to the reward distributions since they also make no assumptions, but can be too conservative in practice.

PAC and uniform-PAC The above bounds on the expected cumulative regret cannot distinguish a policy that makes a few large mistakes or a lot of small mistakes. Depending on the application, either of these might be a preferred quality. If instead of taking expectation, we view the cumulative regret as a random variable, then we can derive high probability bounds. The probably approximately correct (PAC) framework addresses this issue. Very briefly, PAC algorithms are allowed to be approximately accurate with a small probability of failure. Neither PAC nor expected regret guarantees imply convergence to optimal policies with high probability. Dann et al. [2017] provides an extensive discussion of the relationship between these regret bounds, and provides a new notion of uniform-PAC that can imply both PAC and expected regret bounds, but can be harder to achieve.

In conclusion, none of these criteria are perfect by themselves. Worst-case regret is the maximum regret over all possible reward distributions but can be too conservative on easy problems. For the problem-dependent measures, asymptotic optimality can have large burn-in times, and finite-time optimal algorithms might not be asymptotically optimal. While Bayesian regret could be a more realistic measure of performance on average, frequentist analysis provides a more conservative and robust measure of performance. This can be useful in scenarios where the reward distribution is highly uncertain or the consequences of poor performance are severe. Compared to expected regret analysis, PAC and uniform-PAC have high probability guarantees that are potentially harder to achieve.

In the following chapters, we consider the stochastic approach, since it is more robust than the Bayesian

approach but not as conservative as the adversarial approach. For each of our new algorithms, we provided both worst-case and problem-dependent bounds on the expected cumulative regret and also demonstrated their performance on toy examples.

Chapter 3

Exploration in Bandits: O-TS and O-TS$^+$

The fundamental challenge to designing a bandit algorithm is to balance exploitation and exploration. Exploitation maximizes the reward using current knowledge, exploration improves knowledge about the environment at the expense of returns. At each time step, the agent must choose between playing known options with potentially high rewards or exploring unknown options to learn more about the environment and potentially find even better alternatives. One simple exploration policy could be to act greedily and always choose the arm with the highest estimated reward after a fixed period of exploration. However, this policy exploits the best-known option at the expense of further exploring other options, which can lead to sub-optimal results in the long run.

In this chapter, we will discuss a few well-known approaches in multi-arm bandits: epsilon-greedy [Sutton and Barto, 2018], upper confidence bound (UCB) [Auer et al., 2002a] and Thompson sampling (TS) [Agrawal and Goyal, 2017]. The epsilon-greedy algorithm is simple and widely used in practice, but not asymptotically optimal. In contrast, UCB and TS allow agents to explore strategically, taking into account the agent's uncertainty of its estimate. They are proven to achieve asymptotically optimal regret. We will discuss the intuition behind these algorithms as well as the proof sketch, as our proposed algorithms share the same underlying logic and proof techniques.

Table 3.1: Regret bound comparison for O-TS and O-TS$^+$

UCB1 [Auer et al., 2002a]	$\mathbf{O}(\sqrt{KT\ln T})$
TS (Gaussian) [Agrawal and Goyal, 2017]	$\mathbf{O}(\sqrt{KT\ln K})$
O-TS	$\mathbf{O}(\sqrt{KT\ln K})$
O-TS$^+$	$\mathbf{O}(\sqrt{KT\ln T})$

In Section 3.4, we will introduce two other algorithms, Optimistic Thompson Sampling (O-TS), and O-TS$^+$. The O-TS algorithm was first empirically studied by Chapelle and Li [2011]. The key idea for the original O-TS is to clip the posterior distribution optimistically to encourage exploration, while for O-TS$^+$ we clip the distribution even more optimistically at the upper confidence bound. These two algorithms

also achieve near-optimal regret. We will discuss the connections between the regret analysis for these two algorithms and the analysis for UCB and TS, and include an empirical demonstration.

3.1 Epsilon-greedy

Epsilon-greedy is a straightforward method to encourage exploration. Here, the agent plays a random arm with a probability of $\varepsilon \in [0,1]$ (exploration) or the best-known arm with a probability of $1-\varepsilon$ (exploitation). The value of ε is typically chosen to be small, such as 0.1, to ensure that the majority of the time the agent selects the best-known arm. The algorithm is as follows.

Algorithm 3.1 Epsilon-greedy

Input: an arm set $\mathcal{A}$, ε
for round $t = 1, 2, \ldots,$ T **do**
  Draw a number uniformly: $\theta \sim \mathcal{U}(0,1)$
  if $\theta \leq \varepsilon$ **then**
    pull random arm J_t
  else
    pull arm $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \widehat{\mu}_j(t)$
  end if
  Update $\widehat{\mu}_{J_t}$
end for

The main advantage of epsilon-greedy is its simplicity. For this reason, it is widely used in practice. It can work well in scenarios where the reward distributions are relatively stable and the exploration-exploitation trade-off is not critical. Epsilon-greedy also serves as a baseline algorithm for comparison with more sophisticated algorithms that dynamically adjust their exploration strategies.

The main disadvantage of epsilon-greedy is that it requires a fixed value of ε, which may not be optimal for all situations and requires further hyperparameter tuning. Additionally, the epsilon-greedy strategy does not explicitly take into account how uncertain the agent is about the rewards of the different actions. This means it does not guarantee convergence to the optimal solution. The amount of exploration does not change as the agent gathers more data and becomes less uncertain about optimal actions.

There exist more advanced modifications of this algorithm that change ε over time [Tokic, 2010]: In the early stages of the problem when there is limited information about the reward distributions, ε is set to a higher value and the algorithm explores more. As time goes on, ε decreases and the algorithm exploits more, favouring actions that have shown higher historical rewards.

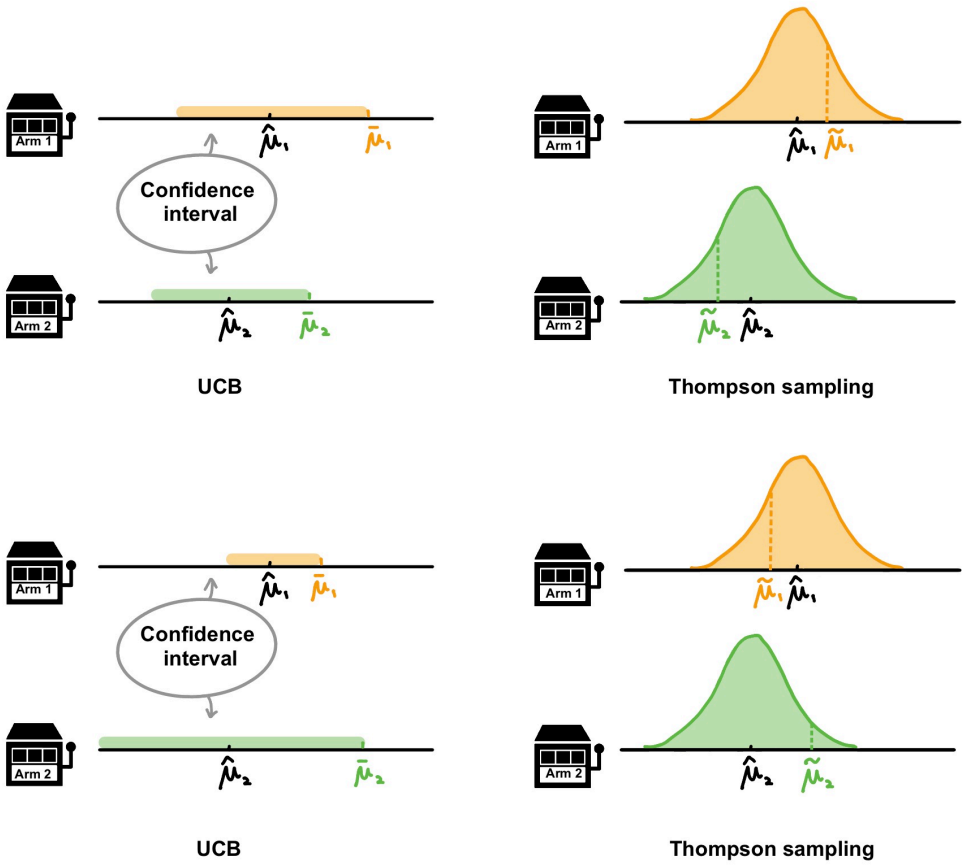

Figure 3.1: UCB and Thompson sampling. $\overline{\mu}$ is the upper confidence bound, and $\widetilde{\mu}$ is a random sample from the posterior distribution. At the top row, arm 1 is chosen in both algorithms (exploitation). At the bottom row, arm 2 is chosen in both algorithms despite having a lower mean reward (exploration).

Table 3.2: Notations for UCB, TS, O-TS and O-TS$^+$

$\widehat{\mu}_{j,O_j(t-1)}$	Empirical mean of arm j
$\overline{\mu}_j(t)$	Upper confidence bound of arm j in round t
$\widetilde{\mu}_j(t)$	Random sample of arm j in round t
$\widetilde{\mu}_j(t)'$	Optimistic random sample of arm j in round t

3.2 Optimism Based Exploration

Optimism-based exploration is based on the principle of *optimism in the face of uncertainty*, which encourages exploration by overestimating the reward of unfamiliar actions. One popular algorithm in this category is upper-confidence bound (UCB) [Auer et al., 2002a].

The UCB algorithm maintains an estimate of the expected reward of each arm, which comes from the average of past rewards of this arm. In addition, it also maintains an upper confidence bound for each arm, which adds a "bonus term" to the empirical reward that quantifies the uncertainty of the estimate. At each round, the UCB algorithm selects the arm with the highest upper confidence bound value. See Algorithm 3.2.

Algorithm 3.2 UCB

Input: an arm set $\mathcal{A}$
Pull each arm once to initialize $O_j, \widehat{\mu}_{j,O_j}$
for round $t = 1, 2, \ldots$ **do**
  for $j \in \mathcal{A}$ **do**
    Set $\overline{\mu}_j \leftarrow \widehat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{2\ln(t)}{O_j(t-1)}}$
  end for
  Pull arm $J_t \leftarrow \arg\max_{j\in\mathcal{A}} \overline{\mu}_j(t)$
  Update $O_{J_t}, \widehat{\mu}_{J_t,O_{J_t}}$
end for

The key idea of the algorithm is that the upper confidence bound is designed using the Hoeffding inequality or other concentration bound such that the true reward of an arm lies within a certain interval with high probability. Intuitively, the bonus term $\sqrt{\frac{2\ln(t)}{O_j(t-1)}}$ decreases as the number of observations of a certain arm increases. As we become more certain about an arm, we will explore it less. We give a brief proof sketch for the regret analysis below.

Concentration inequalities In algorithms for stochastic multi-arm bandit problems, one important step is to estimate the unknown true mean reward of each arm μ_j with the empirical mean reward $\widehat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} X_{J_t}$. Analyzing regret often involves calculating the probability of deviation between the estimate and the true mean, as a function of the number of observations. Concentration inequalities such as Hoeffding's inequality provide a bound on the tail probabilities of random variables. These inequalities are the mathematical foundations in the regret analysis for bandit algorithms.

Lemma 3.2.1 (Hoeffding's Inequality [Hoeffding, 1994])**.** *Let $X_1, \ldots, X_n$ be i.i.d. random variables on $[0,1]$, and let $X = \frac{1}{n}\sum_{i=1}^{n} X_i$, and $\mu = \mathbb{E}[X_i]$. Then, for any $\varepsilon \in \mathbb{R}^+$,*

$$\mathbb{P}(X \geq \mu + \varepsilon) \leq \exp(-2\varepsilon^2 n)$$

$$\mathbb{P}(X \leq \mu - \varepsilon) \leq \exp(-2\varepsilon^2 n)$$

Taking the union bound of the two gives

$$\mathbb{P}(|X - \mu| \geq \varepsilon) \leq 2\exp(-2\varepsilon^2 n)$$

With this tool, we can then analyze the regret of the UCB algorithm.

Theorem 3.2.2 ([Auer et al., 2002a]). *Let $\Delta_j = \mu_1 - \mu_j$. For the K-armed stochastic bandit problem, UCB has problem-dependent regret*

$$Regret_T(UCB) \leq \sum_{j:\Delta_j>0} \frac{8\ln T}{\Delta_j} + (1+\frac{\pi^2}{3})\Delta_j$$

Theorem 3.2.3 ([Auer et al., 2002a]). *For the K-armed stochastic bandit problem, UCB has problem-independent regret*

$$Regret_T(UCB) \leq \mathbf{O}(\sqrt{KT\ln T})$$

Proof sketch for theorem 3.2.2. On a high level, regret occurs when a sub-optimal arm is pulled. It's sufficient to bound $\mathbb{E}[O_i(T)]$, the number of times each sub-optimal arm i is pulled, since

$$\text{Regret}_T(\text{UCB}) = \sum_j \mathbb{E}[O_j(T)]\,\Delta_i$$

.

A sub-optimal arm i can only be selected at round t in three cases:

1. Optimal arm 1 is underestimated: $\widehat{\mu}_1 + \sqrt{\frac{2\ln(t)}{O_1(t-1)}} < \mu_1$
2. Sub-optimal arm j is overestimated: $\widehat{\mu}_j > \mu_j + \sqrt{\frac{2\ln(t)}{O_j(t-1)}}$
3. Sub-optimal arm j is not sufficiently sampled: $O_j(t-1) \leq \frac{8\ln t}{\Delta_j^2}$

Some derivation steps are required to show why these events are a partition for selecting sub-optimal arm j, but we can think about it intuitively with Figure 3.2.

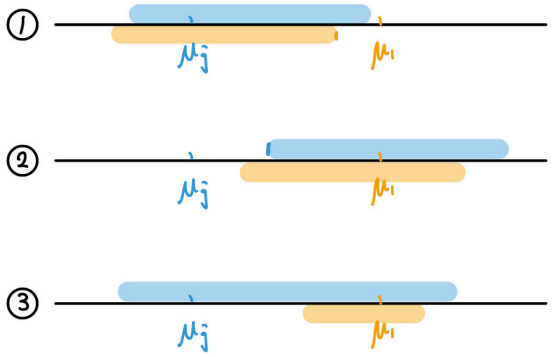

Figure 3.2: In the figure, μ_1 is the true mean of the optimal arm, and μ_j is the true mean of the sub-optimal arm. The yellow and blue highlight bands are confidence intervals respectively. A sub-optimal arm j is pulled if and only if the upper confidence bound of arm j (blue) is above the upper confidence bound of arm 1 (yellow).
In case 1, the confidence interval of arm 1 is entirely below the true mean of arm 1 (underestimation of arm 1). In case 2, the confidence interval of arm j is entirely above the true mean of arm j (overestimation of arm j). In case 3, if arm 1 is not underestimated and arm j is not overestimated, then the confidence interval of arm j must be large enough to cover the difference Δ_j. This can only happen if arm j is not sufficiently sampled and the confidence interval of arm j is not tight.

Then we can upper-bound the first two cases using Hoeffding's inequality. In the first case,

$$\mathbb{P}\left(\widehat{\mu}_1 + \sqrt{\frac{2\ln(t)}{O_1(t-1)}} < \mu_1\right) \leq e^{-2O_1(t-1)\frac{2\ln t}{O_1(t-1)}} = \frac{1}{t^4}$$

And by taking a union bound over the number of times to arm j is pulled in the first t rounds, we can bound this term with a constant. The second case holds with the same argument.

□

Overall, UCB is a powerful and easy-to-implement algorithm with well-understood theoretical guarantees regarding its regret and can achieve near-optimal performance in many scenarios. However, the constant factor of the logarithmic term is not optimal since Hoeffding's inequality cannot be very tight depending on the specific reward distribution.

3.3 Posterior Sampling

Another well-known family of algorithms to balance exploration and exploitation is posterior sampling, also known as Thompson Sampling [Thompson, 1933]. In Thompson Sampling, the agent maintains a posterior probability distribution over the (unknown) mean reward of the unknown reward distribution of each arm, centring around the empirical mean reward from past observation. Agent draws a sample from each posterior distribution at each round and pulls the arm with the highest sampled reward.

The data-dependent posterior distribution takes into account the uncertainty of the agent's estimate of the mean reward of each arm. When the uncertainty is high, the agent will be more likely to explore arms that have not been tried enough, and when the uncertainty is low, the agent will be more likely to exploit the arm with the highest expected reward. The variance of each distribution shrinks as the agent gains more observation of the corresponding arm and becomes more confident about its true mean reward.

Algorithm 3.3 Thompson Sampling with Gaussian priors

Input: an arm set $\mathcal{A}$
Pull each arm once to initialize $O_j, \widehat{\mu}_{j,O_j}$
for round $t = 1, 2, \ldots$ **do**
  for $j \in \mathcal{A}$ **do**
    Draw $\widetilde{\mu}_j(t) \sim \mathcal{N}\left(\widehat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
  end for
  Pull arm $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \widetilde{\mu}_j(t)$
  Update $O_{J_t}, \widehat{\mu}_{J_t,O_{J_t}}$
end for

Intuitively, this algorithm encourages exploration by carefully tuning randomness: The variance term is large enough such that the posterior sample is an overestimate of the true mean reward with a constant probability. In the meantime, it is not too large and decreases with the number of observations. This means that as the agent becomes more confident about a certain arm, the posterior distribution narrows down to the true mean, and in time the agent will more likely pull the optimal arm.

We include a brief proof sketch for the regret analysis for Thompson sampling using Gaussian priors [Agrawal and Goyal, 2017]. The idea is similar to the UCB analysis but relies on concentration inequalities of Gaussian distribution.

Theorem 3.3.1. *Let $\Delta_i = \mu_1 - \mu_i$. For the K-armed stochastic bandit problem, Thompson's sampling with Gaussian prior has problem-dependent regret*

$$R_T(TS) \leq \sum_{j:\Delta_j > 0} \mathbf{O}(\frac{\ln T}{\Delta_j})$$

Theorem 3.3.2. *For the K-armed stochastic bandit problem, Thompson's sampling with Gaussian prior has problem-independent regret*

$$R_T(TS) \leq \mathbf{O}(\sqrt{KT \ln T})$$

Proof sketch for theorem 3.3.1. We want to bound the number of times each sub-optimal arm is pulled, since

$$\mathrm{R}_T(\mathrm{TS}) = \sum_i \mathbb{E}\left[O_i(T)\right] \Delta_i$$

We can then decompose the regret into the following events:

$$\mathcal{A}(t) := \{\widetilde{\mu}_j(t) \leq \mu_j + \frac{1}{2}\Delta_j\}$$

$$\mathcal{B}(t) := \{\widetilde{\mu}_j(t) > \mu_j + \frac{1}{2}\Delta_j\}$$

$$\mathcal{C}(t) := \{O_{t-1}(j) < \mathbf{O}(\frac{\ln(T)}{\Delta_j^2})\}$$

Event $\mathcal{A}$ implies that the optimal arm is underestimated, since $\widetilde{\mu}_1 \leq \widetilde{\mu}_j \leq \mu_j + \frac{1}{2}\Delta_j = \mu_1 - \frac{1}{2}\Delta_j$. Event $\mathcal{B}$ is when sub-optimal arm j is overestimated. Event $\mathcal{C}$ is when sub-optimal arm j is insufficiently sampled, which can be upper bounded by $\mathbf{O}(\frac{\ln(T)}{\Delta_j^2})$.

Event $\mathcal{B}$ can be further separated into two events:

$$\mathcal{B}_1(t) := \{\widetilde{\mu}_j(t) > \mu_j + \frac{1}{2}\Delta_j, |\widehat{\mu}_{j,O_j(t-1)} - \mu_j| \leq \sqrt{\frac{3\ln t}{O_j(t-1)}}\}$$

$$\mathcal{B}_2(t) := \{\widetilde{\mu}_j(t) > \mu_j + \frac{1}{2}\Delta_j, |\widehat{\mu}_{j,O_j(t-1)} - \mu_j| > \sqrt{\frac{3\ln t}{O_j(t-1)}}\}$$

The event $\mathcal{B}_1$ is the posterior deviation. It means that the empirical mean reward is close to the true mean, but the posterior sample is far from the empirical mean. This is a low probability event that can be upper bounded using concentration inequalities of Gaussian distribution.

The event $\mathcal{B}_2$ is the empirical deviation. It means that the empirical mean reward is far from the true mean, in other words, the confidence interval does not hold. This is also a low probability event that can be upper bounded by Hoeffding's inequality when the arm is sufficiently sampled. □

Thompson Sampling has been shown to have strong theoretical guarantees and can perform well in practice. It is more effective in scenarios where the reward distribution is complex or non-parametric, and the variance is difficult to estimate accurately for algorithms such as UCB. However, Thompson Sampling can be computationally expensive since it requires sampling from the posterior distribution for each arm at each time step, whereas UCB only requires calculating an upper confidence bound value for each arm. Additionally, Thompson Sampling can be sensitive to the choice of the prior distribution over the reward.

3.4 Optimistic Thompson Sampling (O-TS), and O-TS$^+$

In our paper [Hu et al., 2023] we propose two other algorithms, O-TS and O-TS$^+$, and provide the regret analysis for them to fill in the gap in relevant bandit theory. The empirical performance of O-TS is mentioned in Chapelle and Li [2011]. Besides enjoying near-optimal regret bounds, the algorithms are also simple to implement and sample/computationally efficient in practice. In the next chapter, we will extend these two algorithms beyond the bandit setting and analyze how they can be extended to reinforcement learning.

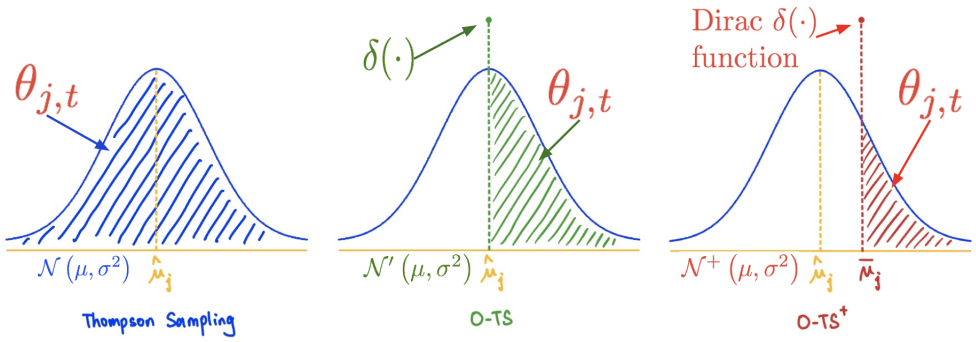

Figure 3.3: Clipping of O-TS and O-TS$^+$, compared to the original TS

Optimistic Thompson Sampling (O-TS), shown in algorithm 3.4, can be viewed as a modified version of Thompson sampling with a modified posterior distribution. Instead of using a full Gaussian distribution to model each arm, we clip each distribution up to the positive half and put a Dirac distribution at the empirical mean. In other words, when we draw a sample $\widetilde{\mu} \sim \mathcal{N}\left(\widehat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$ for arm j at round t, if the sample $\widetilde{\mu}$ is lower than the empirical mean $\widehat{\mu}$, we boost its value to the empirical mean. With probability $\frac{1}{2}$, the sample from this modified distribution will be equal to the empirical mean.

Algorithm 3.4 O-TS for Multi-Armed Bandits

Input: an arm set $\mathcal{A}$
Pull each arm once to initialize $O_j, \widehat{\mu}_{j,O_j}$
for round $t = 1, 2, \ldots$ **do**
  for $j \in \mathcal{A}$ **do**
    Draw $\widetilde{\mu}_j(t) \sim \mathcal{N}\left(\widehat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
    Set $\widetilde{\mu}'_j(t) \leftarrow \max\left\{\widetilde{\mu}_j(t), \widehat{\mu}_j(t)\right\}$
  end for
  Pull arm $J_t \leftarrow \arg\max_{j \in \mathcal{A}} \widetilde{\mu}'_j(t)$
  Update $O_{J_t}, \widehat{\mu}_{J_t,O_{J_t}}$
end for

Clipping the posterior up to the mean makes it more likely for the sample to be optimistic, and therefore encourages exploration while taking variance into account. However, despite its name, O-TS is not an "optimistic-based" algorithm since it only provides weak optimism. The analysis of O-TS is largely similar to Thompson's sampling.

Theorem 3.4.1. *Let $\Delta_i = \mu_1 - \mu_i$. For the K-armed stochastic bandit problem, O-TS has problem-dependent regret*

$$R_T(OTS) \leq \sum_{j:\Delta_j>0} \mathbf{O}\left(\frac{\ln T}{\Delta_j}\right)$$

Proof Sketch. The proof of O-TS is similar to TS. First, We want to bound the number of times each sub-

optimal arm is pulled, since

$$\mathrm{R}_T(\mathrm{TS}) = \sum_i \mathbb{E}\left[O_i(T)\right]\Delta_i$$

We can then define the following events:

$$\mathcal{E}_j^1(t-1) = \left\{ \left| \widehat{\mu}_{j,O_j(t-1)} - \mu_j \right| \leq \sqrt{\frac{0.5\ln\left(T\Delta_j^2\right)}{O_j(t-1)}} \right\}$$

$$\mathcal{E}_j^2(t) = \left\{ \widetilde{\mu}_j'(t) \leq \mu_j + \frac{1}{2}\Delta_j \right\}$$

Event $\mathcal{E}_j^1(t-1)$ is when the confidence interval holds for arm j after the first $t-1$ round. In other words, the empirical mean of arm j is close to the true mean.

Event $\mathcal{E}_j^2(t)$ is when the clipped sample of arm j is not far from the true mean.

For any sub-optimal arm j, let L_j be a positive integer. We will choose the specific number later, but intuitively it denotes the number of observations for arm j such that it is considered "sufficiently sampled". This is important since with enough samples we can apply Hoeffding's inequality to bound tail events of poor estimation.

Then we can decompose the regret using these events:

$$\begin{aligned}
& \mathbb{E}\left[O_j(T)\right] \\
= & \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{J_t = j\right\}\right] \\
\leq & L_j + \sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{J_t = j, O_j(t-1) > L_j\right\}\right] \\
\leq & L_j + \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{J_t = j, \overline{\mathcal{E}_j^1(t-1)}\right\}\right]}_{\mathcal{A}} \\
& + \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{J_t = j, \overline{\mathcal{E}_j^2(t)}, \mathcal{E}_j^1(t-1), O_j(t-1) > L_j\right\}\right]}_{\mathcal{B}} \\
& + \underbrace{\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{1}\left\{J_t = j, \mathcal{E}_j^2(t)\right\}\right]}_{\mathcal{C}}.
\end{aligned} \tag{3.1}$$

Part $\mathcal{A}$ is the empirical deviation. It means that the empirical mean reward is far from the true mean, in other words, the confidence interval does not hold. This is also a low probability event that can be upper bounded by Hoeffding's inequality when the arm is sufficiently sampled.

Part $\mathcal{B}$ is the posterior deviation. It means that the empirical mean reward is close to the true mean, but the posterior sample is far from the empirical mean. This is a low probability event that can be upper bounded using concentration inequalities of Gaussian distribution.

Part $\mathcal{C}$ is when the clipped sample of arm j is close to the true mean, which can be bounded with careful analysis. We refer readers to Theorem 9 in [Hu et al., 2023] and Lemma 2.14 in [Agrawal and Goyal, 2017]. □

We also propose a novel O-TS$^+$ in Algorithm 3.5, an optimistic-based randomized algorithm. Here, we maintain the upper confidence bound of each arm similar to UCB, as well as a posterior distribution similar to Thompson's sampling. Then at each round, we clip the posterior samples up to the upper confidence bound. Different from O-TS, this algorithm is an optimistic algorithm where the clipping is more aggressive.

Algorithm 3.5 O-TS$^+$ for Multi-Armed Bandits

Input: an arm set $\mathcal{A}$
Pull each arm once to initialize $O_j, \widehat{\mu}_{j,O_j}$
for round $t = 1, 2, \ldots$ **do**
  for $j \in \mathcal{A}$ **do**
    Draw $\widetilde{\mu}_j(t) \sim \mathcal{N}\left(\widehat{\mu}_{j,O_j(t-1)}, \frac{1}{O_j(t-1)}\right)$
    Set $\overline{\mu}_j \leftarrow \widehat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\ln(t)}{O_j(t-1)}}$
    Set $\widetilde{\mu}'_j(t) \leftarrow \max\left\{\widetilde{\mu}_j(t), \overline{\mu}_j(t)\right\}$
  end for
  Pull arm $J_t \leftarrow \arg\max_{j\in\mathcal{A}} \widetilde{\mu}'_j(t)$
  Update $O_{J_t}, \widehat{\mu}_{J_t,O_{J_t}}$
end for

The analysis of O-TS$^+$ is very similar to UCB. Here, we will include intuition for the analysis.

Theorem 3.4.2. *Let $\Delta_i = \mu_1 - \mu_i$. For the K-armed stochastic bandit problem, O-TS$^+$ has problem-dependent regret*

$$R_T(OTS^+) \leq \sum_{j:\Delta_j>0} \mathbf{O}(\frac{\ln T}{\Delta_j})$$

Proof Sketch. Regret occurs when a sub-optimal arm is pulled. It's sufficient to bound $\mathbb{E}[O_i(T)]$, the number of times each sub-optimal arm i is pulled:

$$\mathrm{R}_T(\mathrm{OTS}^+) = \sum_j \mathbb{E}[O_j(T)]\,\Delta_i$$

For any sub-optimal arm j, let L_j be a positive integer. We will choose the specific number later, but intuitively it denotes the number of observations for arm j such that it is considered "sufficiently sampled". This is important since with enough samples we can apply Hoeffding's inequality to bound tail events of poor estimation.

We can decompose the regret as follows:

$$\begin{aligned}\mathbb{E}[O_j(T)]\Delta_i &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}(J_t=j)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}(J_t=j, O_j(t-1)\le L_j)\right] + \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}(J_t=j, O_j(t-1)> L_j)\right] \\ &\le L_j + \mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}(J_t=j, O_j(t-1)> L_j)\right]\end{aligned}$$

Then, we just need to bound the event $J_t = j, O_j(t-1) > L_j$. When a sub-optimal arm j is selected at round t while being sufficiently sampled, by contradiction, one of the following must happen:

1. Optimal arm 1 is underestimated: $\widetilde{\mu}_1' < \mu_1$
2. Sub-optimal arm j is overestimated: $\widetilde{\mu}_j' > \mu_j + \sqrt{\frac{24\ln(t)}{O_j(t-1)}}$
3. The upper confidence bound of arm j is not tight, it is above optimal arm: $\sqrt{\frac{24\ln(t)}{O_j(t-1)}} > \Delta_j$

We can upper-bound the first two cases using Hoeffding's inequality. In the first case, since $\widetilde{\mu}_1' \ge \widehat{\mu_1}$, the analysis is exactly the same with UCB:

$$\mathbb{P}\left(\widetilde{\mu}_1' < \mu_1\right) \le \mathbb{P}\left(\widehat{\mu}_1 + \sqrt{\frac{2\ln(t)}{O_1(t-1)}} < \mu_1\right) \le e^{-2O_1(t-1)\frac{2\ln t}{O_1(t-1)}} = \frac{1}{t^4}$$

And by taking a union bound over the number of times to arm j is pulled in the first t rounds, we can bound this term with a constant.

The second case is where the analysis slightly differs from UCB. In addition to Hoeffding's inequality argument, we also need to use the concentration bound of Gaussian distribution to bound the probability of the sample $\widetilde{\mu}_j$ being much larger than the upper confidence bound $\overline{\mu}_j$.

At last, we choose the constant $L_j = \mathbf{O}(\frac{\ln(T)}{\Delta_j^2})$ such that the third case cannot happen, since with enough sample the upper confidence bound should be close to the true mean. Then, we have $\mathbb{E}[O_j(T)] \le \mathbf{O}(\frac{\ln(T)}{\Delta_j^2})$.

Theorem 3.4.3. *For the K-armed stochastic bandit problem, both O-TS and O-TS$^+$ have problem-independent regret*

$$R_T(\textit{O-TS, O-TS}^+) \le \mathbf{O}(\sqrt{KT\ln T})$$

[1]

Proof Sketch. In both proofs for problem-dependent bound for O-TS and O-TS$^+$, we have established that $\mathbb{E}[O_j(T)] \le \mathbf{O}(\frac{\ln(T)}{\Delta_j^2})$.

[1] With more careful analysis, the problem-independent regret bound of O-TS can be further improved to $\mathbf{O}(\sqrt{KT\ln K})$ in our paper. Here we only show the $\mathbf{O}(\sqrt{KT\ln T})$ version.

For a $\Delta \in \mathbb{R}$ that we will choose later, let $\Psi = \{j \in [K] : 0 < \Delta_j < \Delta\}$ to be the set of arms where the sub-optimality gap is smaller than Δ. Let $\overline{\Psi} = \{j \in [K] : \Delta_j \geq \Delta\}$ be the set of arms with larger sub-optimality gap than Δ. The key idea of problem independent bound is to set Δ just large enough such that the regret generated by arms in Ψ dominates the regret generated by arms in $\overline{\Psi}$.

$$\begin{aligned}
R_T(\text{O-TS, O-TS}^+) &= \sum_{t=1}^{T} \mathbb{E}[\mu_1 - \mu_{J_t}] \\
&= \sum_{t=1}^{T} \mathbb{E}[\mu_1 - \mu_{J_t}, \mathbf{1}\{J_t \in \Psi\}] + \sum_{t=1}^{T} \mathbb{E}[\mu_1 - \mu_{J_t}, \mathbf{1}\{J_t \in \overline{\Psi}\}] \\
&\leq T\Delta + \sum_{j \in [K]:\Delta_j \geq \Delta} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}\{J_t = j\}]\Delta_j \\
&= T\Delta + \sum_{j \in [K]:\Delta_j \geq \Delta} \mathbb{E}[O_j(T)]\Delta_j \\
&\leq T\Delta + \sum_{j \in [K]:\Delta_j \geq \Delta} \mathbf{O}(\frac{\ln(T)}{\Delta_j}) \\
&\leq T\Delta + \mathbf{O}(\frac{A\ln(T)}{\Delta})
\end{aligned}$$

Set $\Delta = \sqrt{\frac{A \ln T}{T}}$, then we have

$$R_T(\text{O-TS, O-TS}^+) \leq \mathbf{O}(AT\ln(T))$$

□

3.5 Experiments

In this section, we evaluate the empirical performance of our discussed algorithms O-TS [Chapelle and Li, 2011] and our proposed algorithm O-TS$^+$ for bandits and compared their performance to the UCB [Auer et al., 2002a] and TS [Agrawal and Goyal, 2017] algorithms. We run two sets of experiments shown in Figure 3.4. When an agent pulls an arm, a reward is generated as a Bernoulli random variable with a fixed mean. On the left, we let the mean reward of a 5-armed bandit be $\mu = [0.1, 0.05, 0.03, 0.02, 0.01]$. On the right side, we have a 10-armed bandit with mean $\mu = [0.1, 0.05, 0.05, 0.03, 0.03, 0.02, 0.02, 0.01, 0.01, 0.01]$. We run both for 10^5 rounds and plot the cumulative regret compared to pulling the best arm.

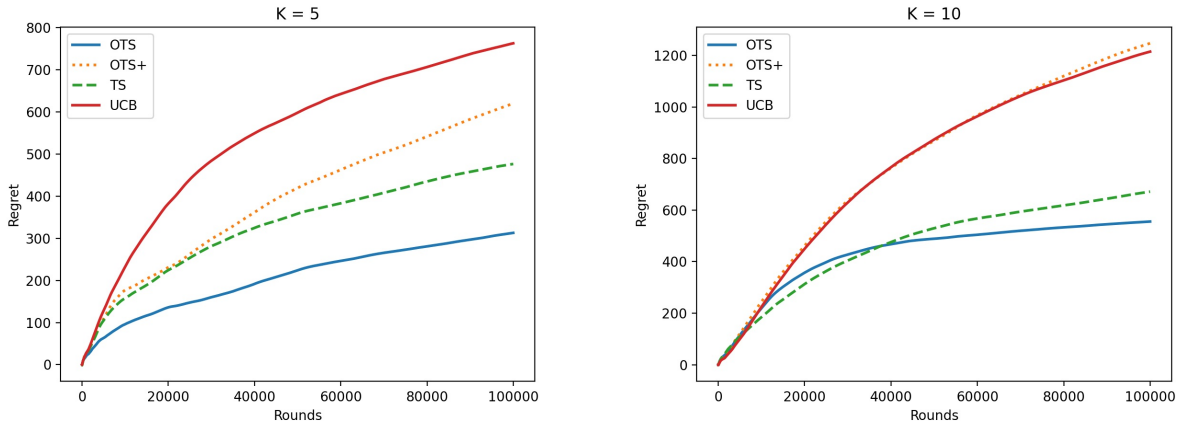

Figure 3.4: Empirical performance for 5 and 10 arms bandits

As shown in Figure 3.4, all algorithm shows a fast increase in regret at the beginning and a slower increase at the end. Thompson sampling tends to perform better than UCB since UCB is overly optimistic. The O-TS algorithm performs similarly to TS and has the lowest cumulative regret in both experiments. This is expected since it is a modified version of TS, and the added optimism prevents underestimation. The O-TS$^+$ algorithm has lower regret than UCB in the 5-armed bandit and performs similarly to UCB in the 10-armed bandit since O-TS$^+$ can be viewed as a randomized version of UCB that is potentially more optimistic.

3.6 Discussion

Optimism-based algorithms are well understood and have been shown to be asymptotically efficient in many settings. Its regret analysis is also more straightforward thanks to the precisely controlled deterministic bonus term. However, the design of the specific upper confidence bound can be challenging to design when the problem is complicated, especially beyond independent arm bandit problems. In contrast, posterior sampling takes advantage of prior knowledge of a specific problem while not compromising on performance.

Our analysis of O-TS and O-TS$^+$ offers insights into the connection between the two and leverages the optimistic but still stochastic samples to boost performance as well as simplify theoretical analysis. These algorithms can be generalized easily to more complicated settings and tend to perform well in practice in both sample and computational efficiency. In the next chapter, we will discuss how the two classes of algorithms generalize to the Markov Decision Process (MDP) algorithms to balance exploration and exploitation in reinforcement learning.

Chapter 4

Markov Decision Process

Reinforcement learning (RL) is a subfield of machine learning that models sequential decision-making. It has a wide range of applications in various fields including robotics [Gu et al., 2017], gaming [Silver et al., 2016], finance [Soleymani and Paquet, 2020], healthcare [Coronato et al., 2020], and transportation [Shalev-Shwartz et al., 2016]. Different from supervised or unsupervised learning, the RL agent's action affects its state in the environment. The goal of RL is to learn a policy that maximizes the cumulative reward obtained by the agent over time.

Reinforcement learning is often modelled using Markov Decision Process (MDP). It has very close connections with the multi-arm bandit problem. Both RL and MAB are types of sequential decision-making problems that estimate the expected reward of each action and rely on a reward signal to guide the agent's behaviour. However, in RL, the agent's action also affects the environment, and there is an extra transition dynamic that takes the agent from one state to another depending on the actions. In other words, a multi-arm bandit can be viewed as a simple MDP with a single state and a set of actions.

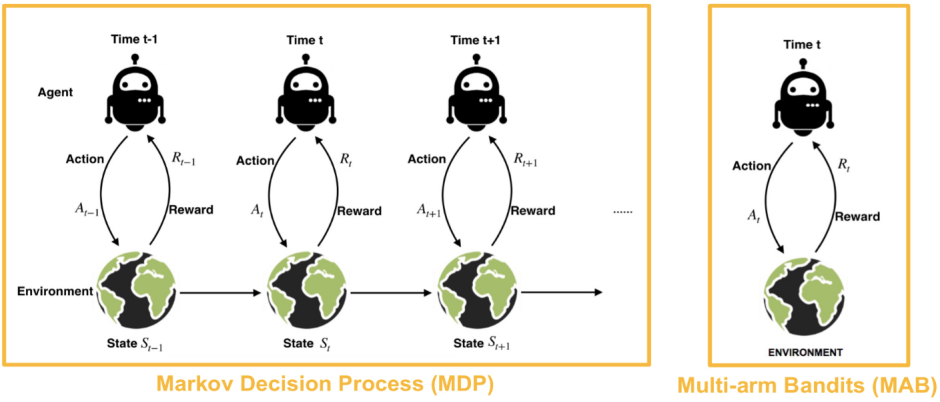

Figure 4.1: Multi-arm bandits can be viewed as stateless MDP [Weng, 2018]

There are several challenges in RL that must be addressed to make it effective and scalable. Similar

to the bandit problem, one main challenge is the exploration-exploitation dilemma: agents must balance exploring the environment to gather new information and exploiting what they already know to maximize rewards. However, this is more difficult in RL compared to bandits, as the environment is more complex with large state and action spaces. The reward attribution to a specific action can be unclear, as sometimes the reward can be delayed throughout the multi-step process.

Another challenge is that data can be expensive to acquire. RL algorithms typically require a large number of interactions with the environment to learn an optimal policy. This can be prohibitively expensive in real-world applications, where data collection is often time-consuming and expensive. A good algorithm should be able to learn without having to observe and memorize all possible state, action and transition dynamics, but rather to explore strategies and make the most use of past experience.

We aim to address both challenges in our work [Hu et al., 2023]. We draw inspiration from the bandit algorithms and study how to extend optimism-based and posterior sampling-based methods to the MDP setting. This allows us to design and analyze RL learning algorithms that tackle the exploration-exploitation dilemma by learning the reward model efficiently from past data. Similar to the bandit analysis, we use regret minimization as our performance measure. Before diving into the details of our algorithm, it's important to clarify the settings and assumptions that we make, as well as some existing related work.

In this chapter, we introduce the definition of MDPs and discuss common assumptions or variations used in literature. Although our theoretical work is done in the finite-horizon tabular setting, they offer insights into the scalability of different approaches. Then, we introduce a few *planning methods*: In bandit algorithms, once we have an estimate of each arm's reward, it is straightforward to pull the arm with the highest reward. In MDPs, this step requires further computation, and an intelligent way of estimation and optimization can lead to much more efficient algorithms. Although computation efficiency is not the main focus, it is still an important step in our algorithms. The most common approach is to maintain a point estimate of the system, including reward and transition dynamics of a certain state and action, or even time in the non-stationary setting. Then, in the finite-horizon algorithms, we could optimize the policy by computing the value function following a backward induction procedure.

After planning, we enter the running phase where the agent acts greedily according to the estimation to collect more samples and then update the estimation. We next include an overview of existing sampling methods in MDP. As inspired by bandit literature, there are also two main approaches: optimism-based, and posterior sampling. We will discuss the differences and connections between the two genres. In the next Chapter, we will discuss how our optimistic Thompson sampling methods in bandit can be extended to MDPs.

4.1 Problem Definition

In this work, we mainly focus on the finite horizon, episodic, non-stationary MDPs defined as follows. We will also briefly discuss these terminologies, and their counterparts such as the infinite horizon, non-episodic, stationary, or continuous setting.

4.1.1 Finite Horizon, Non-stationary, episodic MDP

We consider a Markov Decision Process (MDP) specified by $M = \{\mathcal{S}, \mathcal{A}, \mathcal{H}, \boldsymbol{P}, \boldsymbol{\mu}, p_0\}$, where the notations are defined below:

Table 4.1: Notations for Markov Decision Processes

$\mathcal{S}$	finite state-space with size S
$\mathcal{A}$	finite action-space with size A
$\mathcal{H}$	time horizon (number of rounds in each episode)
$\boldsymbol{P}$	time-dependent transition function
$\boldsymbol{\mu}$	mean of the time-dependent reward function
p_0	initial state distribution
$\pi = (\pi(\cdot,1), ...\pi(\cdot,H))$	Time-dependent policy
$V_t^\pi(s)$	State value function for $s \in \mathcal{S}$
$Q_t^\pi(s,a)$	State-action value function for $s \in \mathcal{S}, a \in \mathcal{A}$
$\mathcal{F}_k = \left\{ s_t^q, a_t^q, X^q_{s_t^q,\, a_t^q, t}, t \in [H], q \in [k] \right\}$	History

An agent interacts with the environment as follows: At the beginning of each episode $k \in [K]$, an initial state s_1^k is drawn from p_0. Then, at each timestep $t \in [\mathcal{H}]$, agent plays an action a_h^k, receives a random reward $X^k_{s_t^k, a_t^k, t} \in [0,1]$ that is drawn from a fixed reward distribution with mean $\mu_{s_t^k, a_t^k, t} = \boldsymbol{\mu}(s_t^k, a_t^k, t)$. Then, the agent transitions into the next state s_{t+1}^k that is sampled from a fixed probability distribution over states, $P_{s_t^k, a_t^k, t} = \boldsymbol{P}(s_t^k, a_t^k, t)$.

Similar to the multi-arm bandit problem, the agent also aims to maximize cumulative reward or minimize regret over a finite number of T episodes, or HT rounds.

A deterministic *policy* $\pi = (\pi(\cdot,1), \pi(\cdot,2), ...\pi(\cdot,H))$ is a sequence of functions where $\pi(\cdot,t) : \mathcal{S} \to \mathcal{A}$ takes a state as input and map it to an action that the agent will take at time t. Let Π be the set of all policies.

A *value function* $V_t^\pi : \mathcal{S} \to \mathbb{R}$ is defined as:

$$V_t^\pi(s) = \mathbb{E}\left[\sum_t^H \mu_{s_t, a_t, t} | \pi, s_t = s \right]$$

where the expectation is taken over the randomness in the transition dynamics. Note that $V_t^\pi(s) \in [0, H-t+1]$, and that $V_1^\pi(s) \geq V_2^\pi(s) \geq ... \geq V_H^\pi(s)$ since rewards are non-negative. Or recursively, define $V_{H+1} = \vec{0}$,

$$V_t^\pi(s) = \mu_{s,\pi(s,t),t} + P^\intercal_{s,\pi(s,t),t} V_{t+1}^\pi$$

Similarly, the *state-action* value function $Q_t^\pi(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q_t^\pi(s,a) = \mathbb{E}\left[\sum_t^H \mu_{s_t,a_t,t} | \pi, s_t = s, a_t = a\right]$$

Notice that $V_t^\pi(s) = Q_t^\pi(s, \pi(s,t))$. Or recursively, let $Q_{H+1}^\pi(a,s) = 0 \; \forall s \in \mathcal{S}, a \in \mathcal{A}, \pi \in \Pi$,

$$Q_t^\pi(s,a) = \mu_{s,a,t} + P_{s,a,t}^\intercal V_{t+1}^\pi = \mu_{s,a,t} + P_{s,a,t}^\intercal Q_{t+1}^\pi(\cdot, \pi(s,t))$$

If we know $\boldsymbol{P}$ and $\boldsymbol{\mu}$, then we can compute the *optimal policy* π_* via backward induction. For $t = H-1, ..., 1$:

$$\pi_*(s,t) = \arg\max_a \left(\mu_{s,a,t} + P_{s,a,t}^\intercal V_{t+1}^{\pi_*}\right) = \arg\max_a Q_t^{\pi_*}(a,s)$$

where

$$V_t^{\pi_*}(s) = \mu_{s,\pi_*(s,t),t} + P_{s,\pi_*(s,t),t}^\intercal V_{t+1}^{\pi_*} = \max_a Q_t^{\pi_*}(a,s)$$

$$Q_t^{\pi_*}(s,a) = \mu_{s,a,t} + P_{s,a,t}^\intercal V_{t+1}^{\pi_*} = \mu_{s,a,t} + P_{s,a,t}^\intercal (\max_a Q_{t+1}^{\pi_*}(\cdot, a))$$

This is referred to as the Bellman optimality equations [Bellman, 1966]. The *history trajectory* drawn by the end of episode k following policies $\pi_1, \ldots, \pi_k$ is defined as follow:

$$\mathcal{F}_k = \left\{ s_t^q, a_t^q, X_{s_t^q,\, a_t^q, t}^q, t \in [H], q \in [k] \right\}$$

where the initial state in each episode is drawn independently from p_0. Define $\mathcal{F}_0 = \{\}$.

The *regret* of a series of policy $\pi_1, \ldots, \pi_k$ over T episodes is defined as:

$$\mathrm{R}_T(\pi_1, ..., \pi_k) = \sum_{k=1}^T \mathbb{E}\left[V_1^{\pi_*}(s_1^k) - V_1^{\pi_k}(s_1^k)\right]$$

where $s_1^k \sim p_0$.

4.1.2 Other Common Settings

As mentioned above, we focus on the finite horizon, episodic, and non-stationary MDPs, as it is simpler to analyze and provide theoretical guarantees. Another common setting that is often preferred in practical RL is the infinite, discounted reward MDPs. Here, the temporal information is incorporated into the states directly. To understand this, let us first discuss a few important concepts:

Episodic vs continues learning Episodic RL is when the agent interacts with the environment for a fixed number of time steps or until it reaches a terminal state. At the end of each episode, the environment is reset to its initial state and a new episode begins. Here, the goal is to maximize the total reward obtained within

each episode. This means that there is a need for strategic exploration in order to maximize reward within a limited number of steps, but the agent can also afford to explore more freely as everything resets at the beginning of the next episode.

There is also the non-episodic setting where the agent interacts with the environment continuously over an indefinite period of time. The agent must balance exploration and exploitation continuously while ensuring that it does not get stuck in a suboptimal policy since there is no sense of rest.

Stationary vs time-dependent (non-stationary) Stationary MDP refers to when the reward structure, transition dynamics, and consequently the policy, all remain fixed over time throughout the learning process. A policy in a stationary environment can be evaluated by averaging the rewards obtained over a fixed number of trials. In contrast, in time-dependent (non-stationary) MDP, the transition probabilities and rewards can be a function of time, and we typically look at cumulative rewards over a fixed time horizon.

In practice, stationary RL is typically easier to learn than non-stationary RL because the environment remains fixed. In non-stationary RL, the agent needs to adapt to changes in the environment and update its policy accordingly. In addition, the policy and values are much larger to store and therefore rarely used in practice. However, much theoretical analysis of RL considers the finite horizon non-stationary scenario. The finite horizon simplifies the analysis, while the non-stationary assumption still allows temporal information, thus approximating the infinite horizon, stationary scenario.

Finite horizon vs infinite horizon The finite horizon is when the agent interacts with the environment for a fixed number of time steps, after which the episode ends. The goal of the agent is to maximize the cumulative reward obtained over the fixed number of time steps. On the other hand, in infinite horizon RL the agent interacts with the environment indefinitely. Typically in the infinite setting, we consider the discounted future reward calculated by summing the rewards obtained by the agent at each time step, discounted by a factor of $\gamma \in [0,1]$ raised to the power of the number of time steps taken to obtain the reward. Mathematically, the value function of policy π in the infinite, stationary, discounted future reward setting is defined as:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s\right]$$

The discount factor γ determines the importance of future rewards relative to immediate rewards. A value of γ close to 0 indicates that the agent is myopic and only cares about immediate rewards, while a value of γ close to 1 indicates that the agent values future rewards almost as much as immediate rewards. By discounting future rewards, the agent takes into account the uncertainty of future rewards, as future rewards are worth less than immediate rewards due to the delay in receiving them.

In finite-horizon, the agent can use dynamic programming to learn an optimal policy. These methods are not applicable in the infinite horizon and the agent must use other techniques such as Monte Carlo methods, temporal difference learning, or function approximation.

Stochastic vs deterministic policy A deterministic policy maps states directly to one action, while a stochastic (randomized) policy assigns a probability distribution over actions for each state. It is shown [Agarwal et al., 2019] [Puterman, 2014] that in the infinite, discounted MDPs, there always exists a stationary and deterministic policy that is optimal. Therefore, the restriction to only using deterministic policies will not affect performance. In our work, the policy is non-stationary since the MDP itself is non-stationary, but we restrict it to be deterministic for simplicity.

Model-based vs model-free Our algorithm is considered to be model-based. Model-based RL involves learning a model of the environment, which represents the transition dynamics between states and actions and the associated reward structure. Once the model is learned, the agent can use it to plan its actions and make optimal decisions. Model-based algorithms typically enjoy good theoretical results and require fewer samples, but can have high computation/memory costs. The choice of model can also impact the robustness and generalization to different environments.

Model-free RL, on the other hand, does not involve learning a model of the environment. Instead, it directly learns an optimal policy through trial-and-error interactions with the environment. Model-free RL algorithms estimate the value function or the policy directly from the experience, without explicitly modelling the transition dynamics. Model-free algorithms typically require a large amount of experience to estimate the value function or policy accurately but are often more efficient computationally and therefore more scalable.

4.2 Planning

Assuming we know all the parameters of an MDP $M = \{\mathcal{S}, \mathcal{A}, \mathcal{H}, \boldsymbol{P}, \boldsymbol{\mu}, p_0\}$, we can compute the optimal policy exactly or approximately.

4.2.1 Backward induction for finite horizon MDPs

In the finite horizon setting exact calculation of the optimal policy can be done using backward induction with a computational cost of $O(AS^2T)$. In our algorithm, the agent cycles between the two phases: sampling and planning. In the sampling phase, the agent acts according to the policy set in the planning phase. The agent collects a data trajectory and adds it to the history. Then in the planning phase, the agent computes the optimal policy under the estimated MDP, which comes from the historical data collected in the sampling phase.

Algorithm 4.1 Backward Induction in Finite Horizon MDP

$V_{H+1} = \vec{0}$
for $t = H, ..., 1$ **do**
  for $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $Q_t(s,a) = \mu_{s,a,t} + P_{s,a,t}^{\mathsf{T}} V_{t+1}$
    end for
    $V_t(s) = \max_{a \in A} Q_t(s,a)$
    $\pi(s,t) = \arg\max_{a \in A} V_t(s)$
  end for
end for

4.2.2 Value and Policy iterations for infinite discounted MDPs

In the infinite horizon setting, there are a few commonly used planning methods. Iterative methods such as value iteration (VI) and policy iteration(PI) are studied extensively in both exact and approximate optimally. Their progress relies on the contraction property of the discounted MDPs. Typically, VI takes more iterations to converge than PI, but each iteration of PI is much more expensive than that of VI. Both methods have also been adjusted to the deep learning settings. With neural network function approximation and incremental update, VI leads to algorithms like DQN [Mnih et al., 2015], and PI leads to things like policy gradient Silver et al. [2014]. In the finite horizon, the analogy of both algorithms is equivalent to backward induction [Agarwal et al., 2019].

Algorithm 4.2 Value Iteration

Input: Discount factor γ, accuracy ε
Initialize $V(s) = 0$ for each state
while $|V(s) - V_{old}(s)| \geq \varepsilon$ for any $s \in S$ **do**
  $V_{old} = V$
  for $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $Q(s,a) = \mu_{s,a} + \gamma P_{s,a}^{\mathsf{T}} V_{old}$
    end for
    $V(s) = \max_{a \in \mathcal{A}} Q(s,a)$
    $\pi(s) = \arg\max_{a \in \mathcal{A}} V(s)$
  end for
end while
return π

Algorithm 4.3 Policy Iteration

Input: Discount factor γ
Initialize π randomly
while $\pi \neq \pi_{old}$ **do**
  $\pi_{old} = \pi$
  Solve the following for $V(s)$:
  $V(s) = \mu_{s,\pi_{old}(s)} + \gamma P_{s,\pi_{old}(s)}^{\mathsf{T}} V$
  for $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $Q(s,a) = \mu_{s,a} + \gamma P_{s,a}^{\mathsf{T}} V_{old}$
    end for
    $\pi(s) = \arg\max_{a \in \mathcal{A}} Q(s,a))$
  end for
end while
return π

4.2.3 Linear programming for infinite discounted MDPs

There is also a non-iterative planning method: linear programming (LP). This method takes advantage of the rich literature of LP and can be solved by algorithms such as the Simplex or Interior Point Method that is more computationally efficient [Ye, 2011]. We include a brief description here for readers to have a better understanding of MDPs.

For a fixed (possibly stochastic) policy π, we can define a visitation measure over states and actions induced by following π after starting at $s0 \sim \rho$.

$$\mu^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}Pr[s_t = s, a_t = a | s_0 \sim \rho]$$

We call μ^{π} the *state-action occupancy* of policy π. Note that μ is a distribution over $S \times A$. If we take a sum over actions, we can get the state occupancy measure of a policy:

$$d^{\pi}(s) = \sum_{a}\mu^{\pi}(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}Pr[s_t = s|s_0] = (1-\gamma)\rho_s + \gamma\sum_{s'}\sum_{a'}P(s|s',a')\mu(s',a')$$

Then we can also describe a policy by $\pi(a|s) = \frac{\mu^{\pi}(s,a)}{d^{\pi}(s)}$. Note that $\langle \mu, r\rangle = (1-\gamma)\mathbb{E}_{s\sim\rho}V_r^{\pi}(s) = (1-\gamma)V_{\rho}^{\pi}$. Then, we can define the following primal and dual forms:

Primal:

$$\begin{array}{ll} \text{minimize} & \langle \rho, v\rangle \\ \text{subject to} & \sum_{s}(I-\gamma P)v \geq r \\ \text{Or equivalently} & v \geq r + \gamma\langle P, v\rangle \end{array}$$

Dual:

$$\begin{array}{ll} \text{maximize} & \frac{1}{1-\gamma}\langle \mu, r\rangle \\ \text{subject to} & \mu \in F_{\rho} \end{array}$$

where the state-action polytope F_ρ is defined as

$$F_{\rho} = \{\mu \in \mathbb{R}^{S\times A} | \sum_{a}(I - \gamma P_a^{\intercal})\mu_a = (1-\gamma)\rho\}$$

Which comes from

$$\forall s, \sum_{a}\mu_{s,a} = \gamma\sum_{a}(P_a^{\intercal}\mu_a)_s + (1-\gamma)\rho(s)$$

or

$$\sum_{a}\mu_a - \gamma P_a^{\intercal}\mu_a = (1-\gamma)\rho$$

since this is the set of linear constraints such that $\mu \in F_\rho$ if and only if there exists a stationary (and possibly randomized) policy π such that $\mu_{\rho}^{\pi} = \mu$. Notice that in the primal form, $v \geq r + \gamma\langle P, v\rangle$ is the Bellman optimality equation when $v = v*$.

4.3 Sampling Algorithms in Finite Horizon MDPs

Statistical efficiency is one of the main challenges in RL as data collection in real life can be expensive, and the state space can be intractably large. It is especially challenging to explore strategically in RL since often rewards can be delayed. An action might not be immediately optimal, but can potentially be beneficial in the future by being on the right trajectory. It is not always obvious which action or trajectory yields more cumulative reward, or will be more informative in exploration. The need for an intelligent exploration strategy is therefore crucial.

Bandit literature offers various approaches that one can adapt to MDPs. A widely used approach in MDPs is also ε-greedy, where the majority of the time the agent follows the optimal trajectory in its experience, and with some small probability the agent explores random actions. Despite the simple implementation, this method takes exponentially long to learn [Osband et al., 2019]. This is because the algorithm does not prioritize exploring the more informative states, and the proportion of exploration stays fixed throughout the learning process. Improved methods take into account the agent's uncertainty toward any trajectory while also aiming to be efficient statistically and computationally.

Optimism-Based Following the principle of *optimisim in the face of uncertainty*, much work extends the UCB-styled algorithms from bandits to MDPs. At the beginning of each episode, optimism-based algorithms construct a carefully calculated upper confidence bound to represent a likely yet optimistic outcome based on historical observations. The confidence upper bound is constructed by adding a bonus term to the empirical estimate. The agent then follows a greedy policy with respect to this deterministic upper bound.

The key idea is to be overly optimistic toward unfamiliar states and actions to encourage the exploration of poorly understood trajectories. The bonus term is inversely proportional to the square root of the number of times this action has been taken in a certain state and at a certain time. This is to increase the probability of the agent playing the less familiar actions (exploration). Nevertheless, the bonus term will shrink as the agent collects more data to allow convergence to optimality.

There are many UCB-based algorithms in the RL literature. Azar et al. [2017] proposed Upper Confidence Bound Value Ieration (UCB-VI), a modified version of value iteration with an exploration bonus. It enjoys the (near)-optimal $\widetilde{O}\left(\sqrt{ASH^3T}\right)$ regret bound up to a logarithmic factor. Dann et al. [2017] proposed the "Upper Bounding Expected next state Value (UBEV)" algorithm and provided uniform high-probability regret analysis. Zhang et al. [2020] provides a model-free algorithm UCB-Advantage and proves that it achieves near-optimal regret bound. Tiapkin et al. [2022] propose the Bayes-UCBVI algorithm that uses the quantile of a Q-value function posterior as the upper confidence bound on the optimal Q-value function. Their algorithm is also proven to be near optimal. Despite the abundant theoretical analysis, the optimism-based methods can be computationally expensive and often do not perform as well as posterior sampling in practice [Osband and Roy, 2017].

Posterior sampling-Based On the other hand, the posterior sampling-based method encourages exploration by randomizing the obtained data. Because of the randomization, the learning agent is encouraged to visit states and actions that have been less explored. Randomness can be achieved by injecting noise into the data. For example, many related methods add calibrated Gaussian noise to the empirical estimates of the rewards. A model-based algorithm, NARL-UCBVI [Pacchiano et al., 2021] and a model-free algorithm, C-RLSVI [Agrawal et al., 2020] achieve the same $\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$ regret bound. Very recently, another model-free algorithm SSR-Bernstein [Xiong et al., 2022] tightens the regret bound to $\widetilde{O}\left(\sqrt{ASH^3T}\right)$. Although the transition dynamics are also unknown, these learning algorithms do not add any noise to the empirical estimates of the transition probability distributions. Empirically, Osband et al. [2019] has shown that learning algorithms with random noise can explore efficiently.

One could also extend the Thompsom sampling algorithm from bandit to MDPs more directly. These algorithms maintain a data-dependent distribution, often with a carefully designed variance to account for uncertainty. For example, Gaussian distributions can be used to model the means of the reward distributions and Dirichlet distributions can be used to model the transition probability distributions in MDPs. In fact, the exact posterior distribution is not required explicitly, so long as random samples can be drawn. The agent then acts greedily and plays the action with the highest random sample. Chapelle and Li [2011] studied the practical performance of Thompson Sampling extensively. The theoretical performance of Thompson Sampling depends on the choice of the prior distributions. Thompson Sampling with Beta priors is asymptotically optimal [Agrawal and Goyal, 2017, Kaufmann et al., 2012], while Thompson Sampling with Gaussian priors is problem-dependent optimal [Agrawal and Goyal, 2017]. Limitations of existing posterior sampling-based methods include complicated theoretical analysis and computational efficiency: algorithms often need to draw more samples than necessary to construct a model.

Algorithm 4.4 Optimism-based RL

Input: History buffer, bonus constructor
for episode $k = 1, 2, ..., K$ **do**
  Planning:
  for step $t = 1, 2, ..., H$ **do**
    Compute bonus term for each (s, a) based on history
    Construct UCB and compute Q_t^k
  end for
  Sampling:
  for step $t = 1, 2, ..., H$ **do**
    Take action $a_t^k = \arg\max_a Q_t^k(s, a)$
    Update history buffer
  end for
end for

Algorithm 4.5 Posterior Sampling in RL

Input: History buffer, bonus constructor
for episode $k = 1, 2, ..., K$ **do**
  Planning:
  for step $t = 1, 2, ..., H$ **do**
    Construct posterior distribution for each (s, a) based on history
    Sample from distribution, compute Q_t^k
  end for
  Sampling:
  for step $t = 1, 2, ..., H$ **do**
    Take action $a_t^k = \arg\max_a Q_t^k(s, a)$
    Update history buffer
  end for
end for

As summarized in Osband and Roy [2017], there are many connections and similarities between the two algorithms 4.5 and 4.4. Osband et al. [2016b] compared the two types of algorithms, and argued that the posterior sampling-based method can be viewed as stochastically optimistic. They argue that despite the simplicity, the existing optimism-based algorithms do not enjoy the same statistical and computational efficiency compared to posterior sampling. The reason is that the deterministic confidence sets constructed in UCB-like algorithms are often not optimal. In the next Chapter, we will introduce our algorithms that modify the posterior distribution to be more optimistic, with the hope of achieving the best of both worlds.

Chapter 5

Optimistic Thompson Sampling in MDPs

As discussed in the previous chapters, Thompson sampling maintains data-dependent distributions to estimate parameters of the environment (bandits or MDPs). The agent acts greedily with respect to random samples at each round, and the resulting policy is shown to balance the exploration-exploitation trade-off. The Gaussian distribution, thanks to its well-developed theory and nice properties, became a popular choice of prior distribution used to model the reward [Agrawal and Goyal, 2017].

However, one does not need to use the exact Gaussian distribution to be able to take advantage of these nice properties. In fact, some recent work in bandits demonstrated the benefits of reshaping the posterior distribution in various ways. For stochastic bandits, MOTS Jin et al. [2021] reshapes the posterior distribution by clipping the upper tail and boosting the variance of the posterior distribution. It is the first Thompson Sampling-like algorithm to achieve minimax optimality. Later, Hu and Hegde [2022] reshapes the posterior distribution by adding a bonus term to the mean to devise an optimal Thompson Sampling-like algorithm in the context of differentially private stochastic bandits.

In this chapter, we discuss two novel algorithms proposed in our work [Hu et al., 2023]: O-TS-MDP, and O-TS-MDP$^+$. The two algorithms can be viewed as Thompson sampling-like learning algorithms with posterior distribution reshaping. They are extensions of our previous bandit algorithms O-TS and O-TS$^+$ to the MDP setting: they modify the posterior distributions of the reward in a more optimistic way and therefore may improve the performance as well as simplify the analysis.

Table 5.1: Regret bound comparison for O-TS-MDP and O-TS-MDP$^+$

UCB-VI [Dann et al., 2017]	$\widetilde{O}\left(\sqrt{ASH^3T}\right)$	Model-based	Deterministic
RLSVI [Russo, 2019]	$\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$	Model-free	Randomized
O-TS-MDP	$\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$	Model-based	Randomized
O-TS-MDP$^+$	$\widetilde{O}\left(\sqrt{ASH^3T}\right)$	Model-based	Randomized

Our first algorithm, Optimistic Thompson Sampling for MDPs (O-TS-MDP), achieves a $\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$

regret bound, where S is the size of the state space, A is the size of the action space, H is the number of time-steps per episode and T is the number of episodes. Our second algorithm, Optimistic Thompson Sampling plus for MDPs (O-TS-MDP$^+$), achieves the (near)-optimal $\widetilde{O}\left(\sqrt{ASH^3T}\right)$ regret bound by taking a more aggressive clipping strategy.

We will describe the algorithms and the intuitions for their improved regret bounds and performance, and refer readers to the paper [Hu et al., 2023] for detailed proofs. Then, we demonstrate the empirical performance on a simple random MDP and compare it to existing optimism-based and posterior sampling-based algorithms.

We define some additional notations used in our algorithm in Table 5.2. Variance of the Gaussian is choosen to be $\sigma^k_{s,a,t} := 5\sqrt{H^2 \log^2\left(\frac{H}{\delta}\right) / \widehat{O}^{k-1}_{s,a,t}}$, where $\delta = \frac{1}{ASH^2T^2}$. Notice that the number of observations is in the denominator, meaning that with more observations we are more confident about the empirical estimate of a (s,a,t) pair, and therefore have a more concentrated posterior distribution. For the case when (s,a,t) has not been visited yet by the end of episode $k-1$ and $\widehat{O}^{k-1}_{s,a,t} = 0$, our algorithms set $\sigma^k_{s,a,t}$ to a large constant. This is to encourage exploration of the more uncertain state, as a more positive sample is likely to be drawn with a large variance.

Table 5.2: Additional notations for O-TS-MDP and O-TS-MDP$^+$

$\widehat{O}^k_{s,a,t} = \sum_{q=1}^k \mathbf{1}\left\{\left(s_t^q, a_t^q\right) = (s,a)\right\}$	# visitation of (s,a,t) by the end of episode k
$\widehat{\mu}^k_{s,a,t} = \frac{1}{\widehat{O}^k_{s,a,t}} \sum_{q=1}^k \mathbf{1}\left\{\left(s_t^q, a_t^q\right) = (s,a)\right\} X^q_{s,a,t}$	Empirical mean of (s,a,t) by the end of episode k
$\widehat{P}^k_{s,a,t}(s') = \frac{1}{\widehat{O}^k_{s,a,t}} \sum_{q=1}^k \mathbf{1}\left\{\left(s_t^q, a_t^q\right) = (s,a), s^q_{t+1} = s'\right\}$	Empirical transition probability from (s,a,t) to s'
$\sigma^k_{s,a,t}$	Variance of the Gaussian distribution
$\widetilde{\mu}^k_{s,a,t}$	Random sample drawn from unclipped Gaussian for (s,a,t) at episode k
$\widetilde{\mu'}^k_{s,a,t} := \max\left\{\widehat{\mu}^{k-1}_{s,a,t}, \widetilde{\mu}^k_{s,a,t}\right\}$	Optimistic sample used in O-TS-MDP
$\overline{\mu}^k_{s,a,t}$	Upper confidence bound for (s,a,t) after k episode
$\widetilde{\mu'}^k_{s,a,t} = \max\left\{\widetilde{\mu}^k_{s,a,t}, \overline{\mu}^k_{s,a,t}\right\}$	More optimistic sample used in O-TS-MDP$^+$

5.1 OTS-MDP

We propose Optimistic Thomson sampling for MDPs (O-TS-MDP) shown in Algorithm 5.1. In the planning phase, O-TS-MDP uses Gaussian distributions with increased variance centred around the empirical mean reward to model the performance of a state-action-time tuple. When the random posterior sample is smaller than the mean, O-TS-MDP increases the value of the mean reward. In other words, the actual distribution used is a clipped Gaussian with all the mass of the lower half concentrated at the mean as a Dirac distribution. This idea is from the O-TS [Chapelle and Li, 2011] algorithm in bandits mentioned in Chapter 3.

O-TS-MDP follows a similar structure to other model-based algorithms. In episode k, we construct an episode-dependent model $\tilde{M}'_k$ to simulate the true model. To do this, we draws a random sample $\widetilde{\mu}^k_{s,a,t} \sim$

$\mathcal{N}\left(\widehat{\mu}_{s,a,t}^{k-1}, SH\left(\sigma_{s,a,t}^{k}\right)^2\right)$ for each (s,a,t). The clipping is done in the following way: If $\widetilde{\mu}_{s,a,t}^{k} < \widehat{\mu}_{s,a,t}^{k-1}$, we boosts it to $\widehat{\mu}_{s,a,t}^{k-1}$. Our final sample used by the algorithm is $\widetilde{\mu'}_{s,a,t}^{k} := \max\left\{\widehat{\mu}_{s,a,t}^{k-1}, \widetilde{\mu}_{s,a,t}^{k}\right\}$. In other words, $\widetilde{\mu'}_{s,a,t}^{k}$ can be viewed as a random variable drawn from distribution $\mathcal{N'}_{s,a,t}^{k}$ with the following probability density function (PDF):

$$f'(x) = \begin{cases} 0, & x < \widehat{\mu}_{s,a,t}^{k-1}, \\ \phi\left(x; \widehat{\mu}_{s,a,t}^{k-1}, SH\left(\sigma_{s,a,t}^{k}\right)^2\right) + \frac{\delta\left(x - \widehat{\mu}_{s,a,t}^{k-1}\right)}{2}, & x \geq \widehat{\mu}_{s,a,t}^{k-1}, \end{cases}$$

where $\phi(x;\mu,\sigma^2)$ denotes the PDF of $\mathcal{N}\left(\mu,\sigma^2\right)$ and $\delta(\cdot)$ denotes the Dirac delta function.

With the optimistic sample $\widetilde{\mu'}_{s,a,t}^{k}$ for all (s,a,t) in hand, we construct $\tilde{M}'_k = \left\{\mathcal{S}, \mathcal{A}, H, \widehat{P}^{k-1}, \widetilde{\mu'}^{k}, p_0\right\}$, where $\widehat{P}^{k-1} = \left\{\widehat{P}_{s,a,t}^{k-1}\right\}$ collects all the empirical transition probability distributions by the end of episode $k-1$ and $\widetilde{\mu'}^{k} = \left\{\widetilde{\mu'}_{s,a,t}^{k}\right\}$ collects all the random samples after the boosting. After constructing $\tilde{M}'_k$, O-TS-MDP uses backwards induction to find the optimal policy π_k for $\tilde{M}'_k$ (shown in Line 4 to Line 12 in Algorithm 5.1). Let $\widetilde{V'}_t^{\pi}$ denote the value functions of a fixed policy π for $\tilde{M}'_k$ in round t.

Algorithm 5.1 O-TS-MDP

1: **Input:** MDP instance M, number of episodes T
2: **Initialization:**
Set $\widehat{O}_{s,a,t} \leftarrow 0$, $\widehat{P}_{s,a,t} \leftarrow \vec{0}$, $\widehat{\mu}_{s,a,t} \leftarrow 0, \forall (s,a,t)$
3: **for** episode $k = 1, 2, \ldots, T$ **do**
4: Set $\widetilde{V'}_{H+1}^{\pi_k} = \vec{0}$
5: **for** $t = H, H-1, \ldots, 1$ **do**
6: **for** $s \in \mathcal{S}$ **do**
7: **for** $a \in \mathcal{A}$ **do**
8: Draw $\widetilde{\mu}_{s,a,t} \sim \mathcal{N}\left(\widehat{\mu}_{s,a,t}, \left(\sqrt{SH}\sigma_{s,a,t}^{k}\right)^2\right)$
Set $\widetilde{\mu}'_{s,a,t} \leftarrow \max\left\{\widetilde{\mu}_{s,a,t}, \widehat{\mu}_{s,a,t}\right\}$
Set $\widetilde{Q}_{s,a,t} \leftarrow \widetilde{\mu}'_{s,a,t} + \widehat{P}_{s,a,t}^{\intercal} \widetilde{V'}_{t+1}^{\pi_k}$
9: **end for**
10: Set $\pi_k(s,t) \leftarrow \arg\max_{a \in \mathcal{A}} \widetilde{Q}_{s,a,t}$
Set $\widetilde{V'}_t^{\pi_k}(s) \leftarrow \widetilde{Q}_{s,\pi_k(s,t),t}$
11: **end for**
12: **end for**
13: Sample $s_1^k \sim p_0$, run π_k, and update $\widehat{\mu}_{s_t^k,\pi_k(s_t^k,t),t}$, $\widehat{O}_{s_t^k,\pi_k(s_t^k,t),t}$, and $\widehat{P}_{s_t^k,\pi_k(s_t^k,t),t}$ for all $t \in [H]$.
14: **end for**

O-TS-MDP is both computationally efficient and space efficient and enjoys elegant theoretical analysis. At each episode, O-TS-MDP only draws one random sample for each state-action-time tuple and enjoys a

$\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$ regret bound. Per episode, the time complexity is $O(AS^2H)$ and the space complexity is $O(AS^2H)$. Although the regret bound of O-TS-MDP is not as tight as OPSRL [Tiapkin et al., 2022] and SSR-Bernstein [Xiong et al., 2022], OPSRL needs to draw $\widetilde{O}(1)$ random samples while SSR-Bernstein is a model-free algorithm and therefore inherently different.

Intuitively, clipping the lower half of the Gaussian distribution guarantees that the sampled parameter is at least as good as the empirical, which encourages the exploration of the uncertain. Theoretically, the reshaping of the posterior distribution plays a crucial role in simplifying the algorithm and the analysis. The one-sided distribution helps to avoid upper bounding the absolute value of the estimation error, thus simplifying the theoretical analysis as compared to the analysis of RLSVI-based algorithms [Agrawal et al., 2020, Russo, 2019, Xiong et al., 2022]. As for the transition kernel, OPSRL and SOS-OPS-RL use Dirichlet random variables to construct the model.

Our methods only model the reward with the posterior distributions. Different from stochastic bandits where the rewards are the only unknown factor, in MDPs the transition dynamics are also unknown. In our work, we directly use the empirical estimate for the transition dynamics for simplicity, since the reward randomization in our algorithm is enough to drive exploration in the analysis. Two Thompson Sampling-like algorithms, SOS-OPS-RL [Agrawal et al., 2020] and OPSRL [Tiapkin et al., 2022], use Dirichlet distributions to model the transition probability distributions. They boost the variance of the Dirichlet distributions to encourage exploration. SOS-OPS-RL achieves a $\widetilde{O}\left(\sqrt{AS^2H^4T}\right)$ regret bound while SPSRL achieves the (near)-optimal $\widetilde{O}\left(\sqrt{ASH^3T}\right)$ regret bound.

5.2 O-TS-MDP$^+$

The O-TS-MDP$^+$ Algorithm 5.2 is similar to O-TS-MDP but uses a more aggressive clipping for the Gaussian distribution. Different from O-TS-MDP, O-TS-MDP$^+$ is an optimism-based algorithm with randomization. O-TS-MDP$^+$ boosts the random sample to the upper confidence bound if smaller than the upper confidence bound. This aggressive clipping strategy contributes to reducing the variance of the posterior distribution as compared to O-TS-MDP, which, consequently, leads to tightening the regret bound to $\widetilde{O}\left(\sqrt{ASH^3T}\right)$.

Similar to Algorithm 5.1, in each episode, O-TS-MDP$^+$ also constructs a model $\tilde{M}'_k = \left\{\mathcal{S}, \mathcal{A}, H, \widehat{P}^{k-1}, \widetilde{\mu'}^k, p_0\right\}$. The only difference is in how to construct $\widetilde{\mu'}^k$. Let $\overline{\mu}^k_{s,a,t} := \widehat{\mu}^{k-1}_{s,a,t} + 2\sigma^k_{s,a,t}$ be the upper confidence bound for (s,a,t) which is determined by $\mathcal{F}_{k-1}$. At the beginning of episode k, for each (s,a,t), O-TS-MDP$^+$ draws a random sample $\widetilde{\mu}^k_{s,a,t} \sim \mathcal{N}\left(\widehat{\mu}^{k-1}_{s,a,t}, \left(\sigma^k_{s,a,t}\right)^2\right)$. Then, O-TS-MDP$^+$ boosts it to $\overline{\mu}^k_{s,a,t}$ if $\widetilde{\mu}^k_{s,a,t} < \overline{\mu}^k_{s,a,t}$. The final sample used in the model is $\widetilde{\mu'}^k_{s,a,t} = \max\left\{\widetilde{\mu}^k_{s,a,t}, \overline{\mu}^k_{s,a,t}\right\}$ denote the sample value after the boosting. The PDF for the distribution of $\widetilde{\mu'}^k_{s,a,t}$ can be defined as follow:

$$f'(x) = \begin{cases} 0, & x < \overline{\mu}^k_{s,a,t}, \\ \phi\left(x; \widehat{\mu}^{k-1}_{s,a,t}, (\sigma_{s,a,t})^2\right) + \Phi\left(\overline{\mu}^k_{s,a,t}; \widehat{\mu}^{k-1}_{s,a,t}, (\sigma_{s,a,t})^2\right)\delta(x - \overline{\mu}^k_{s,a,t}), & x \geq \overline{\mu}^k_{s,a,t}, \end{cases}$$

where $\Phi(x;\mu,\sigma^2)$ denotes the cumulative distribution function (CDF) of $\mathcal{N}\left(\mu,\sigma^2\right)$.

Let $\widetilde{\mu'}^k = \left\{\widetilde{\mu}'^k_{s,a,t}\right\}$ collect all the samples after the boosting. After constructing $\tilde{M}'_k$, O-TS-MDP$^+$ computes the optimal policy π_k for $\tilde{M}'_k$ by using backwards induction. Algorithm 5.2 presents the pseudo-code of O-TS-MDP$^+$. The differences between O-TS-MDP and O-TS-MDP$^+$ are highlighted in Algorithm 5.1 and Algorithm 5.2, respectively.

Algorithm 5.2 O-TS-MDP$^+$

Input: MDP instance M, number of episodes T
Initialization:
  Set $\widehat{O}_{s,a,t} \leftarrow 0$, $\widehat{P}_{s,a,t} \leftarrow \vec{0}$, $\widehat{\mu}_{s,a,t} \leftarrow 0, \forall(s,a,t)$
for episode $k = 1,2,\ldots,T$ **do**
  Set $\widetilde{V}'^{\pi_k}_{H+1} = \vec{0}$
  for $t = H, H-1, \ldots, 1$ **do**
    for $s \in \mathcal{S}$ **do**
      for $a \in \mathcal{A}$ **do**
        Draw $\widetilde{\mu}_{s,a,t} \sim \mathcal{N}\left(\widehat{\mu}_{s,a,t}, \left(\sigma^k_{s,a,t}\right)^2\right)$
        Set $\overline{\mu}_{s,a,t} \leftarrow \widehat{\mu}_{s,a,t} + 2\sigma^k_{s,a,t}$
        Set $\widetilde{\mu}'_{s,a,t} \leftarrow \max\left\{\widetilde{\mu}_{s,a,t}, \overline{\mu}_{s,a,t}\right\}$
        Set $\widetilde{Q}_{s,a,t} \leftarrow \widetilde{\mu}'_{s,a,t} + \widehat{P}^{\intercal}_{s,a,t}\widetilde{V}'^{\pi_k}_{t+1}$
      end for
      Set $\pi_k(s,t) \leftarrow \arg\max_{a\in\mathcal{A}} \widetilde{Q}_{s,a,t}$
      Set $\widetilde{V}'^{\pi_k}_t(s) \leftarrow \widetilde{Q}_{s,\pi_k(s,t),t}$
    end for
  end for
  Sample $s^k_1 \sim p_0$, run π_k, and update $\widehat{\mu}_{s^k_t,\pi_k(s^k_t,t),t}$, $\widehat{O}_{s^k_t,\pi_k(s^k_t,t),t}$ and $\widehat{P}_{s^k_t,\pi_k(s^k_t,t),t}$ for all $t \in [H]$.
end for

O-TS-MDP$^+$ is a model-based, optimistic algorithm, meaning the regret can be decomposed in a certain way. More specifically, if the Optimism Decomposition [Pacchiano et al., 2021] can be used to decompose the regret. The Optimism Decomposition requires the parameters in the constructed model to be greater than the parameters in the true model. Meanwhile, the estimation errors should be maintained low enough.

O-TS-MDP$^+$ achieves the (near)-optimal $\widetilde{O}\left(\sqrt{ASH^3T}\right)$ regret bound. The aggressive clipping contributes to reducing the variance of the reshaped posterior distribution as compared to O-TS-MDP. Consequently, the regret bound is tightened to be (near)-optimal, same as OPSRL of Tiapkin et al. [2022] and SSR-Bernstein of Xiong et al. [2022]. O-TS-MDP$^+$ can be viewed as a randomized version of UCB-VI [Azar et al., 2017].

5.3 Experiments

In this section, we evaluate the empirical performance of our proposed algorithms O-TS-MDP and O-TS-MDP$^+$ for MDPs with $S = [5, 20, 50]$, $A = 3$ and $H = 10$. For a fair performance comparison, our experimental set-up is fully adopted from Dann et al. [2017]. For a specific (s,a,t), the random reward $X_{s,a,t}$ in each episode is drawn from a Bernoulli distribution with parameter $\mu_{s,a,t}$. We set $\mu_{s,a,t} = 0$ with probability 0.85 and with probability 0.15, the value of $\mu_{s,a,t}$ is drawn from a uniform distribution. Similarly, the transition probability distribution $P_{s,a,t}$ is sampled from a Dirichlet distribution with parameters $(0.1, 0.1, \ldots, 0.1)$, meaning that with a high chance the transition probability distribution is concentrated on a single state. The sparsity design is to control the occurrence that sub-optimal policies can obtain rewards by chance.

We compare O-TS-MDP, O-TS-MDP$^+$, SSR-Bernstein [Xiong et al., 2022], and TS-MDP, a Thompson Sampling-based learning algorithm without clipping the posterior distributions, meaning the episode-dependent model are constructed as $\tilde{M} = \left\{\mathcal{S}, \mathcal{A}, H, \widehat{P}^{k-1}, \widetilde{\mu}^k, p_0\right\}$. We set $T = 10^7$ and compare the cumulative average rewards of each episode.

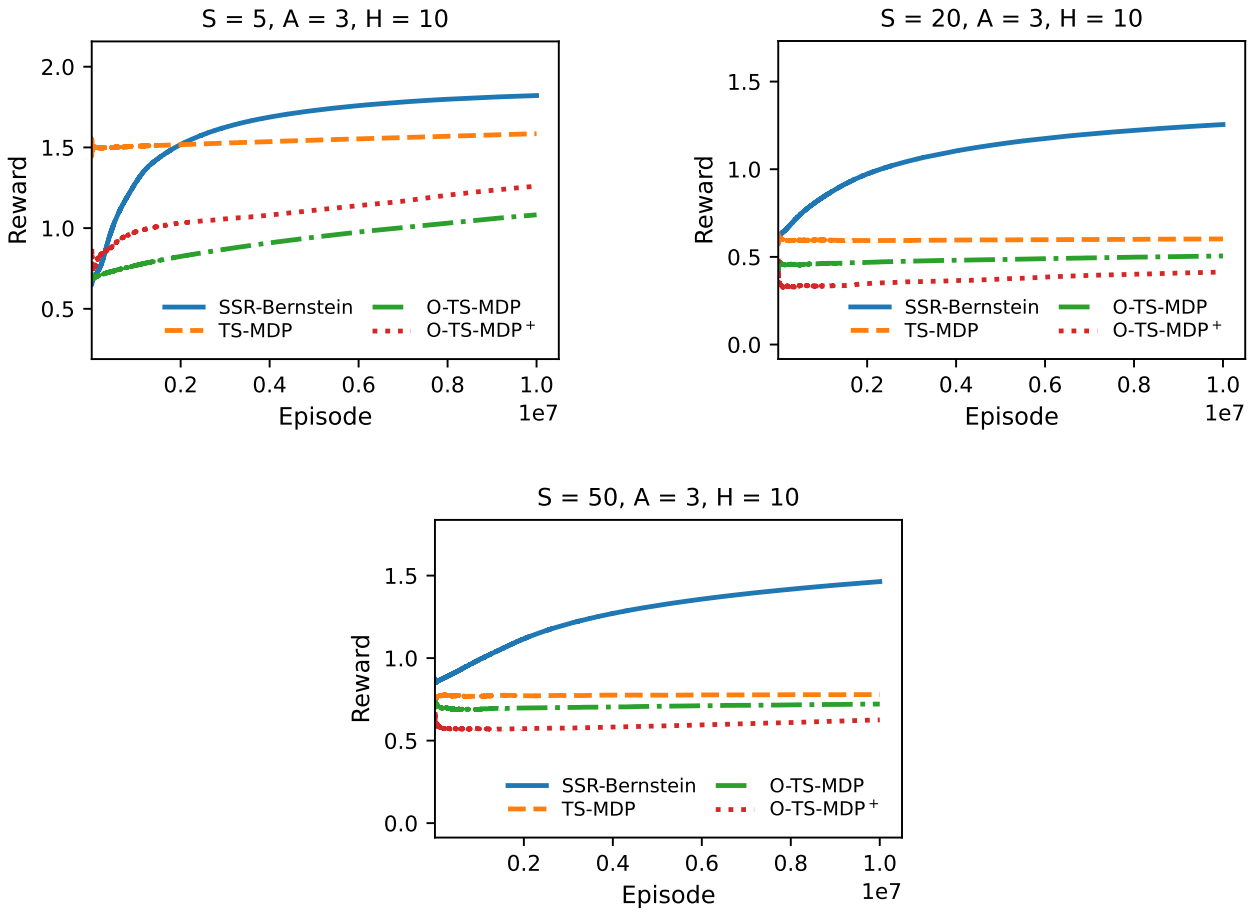

Figure 5.1: Empirical performance for 5, 20 and 50 states

As shown in Figure 5.1, the rewards for all algorithms steadily increase as the learning agent gains a better estimation of the parameters of the true MDP over time. When the number of states is small ($S = 5$), O-TS-MDP$^+$ performs slightly better than O-TS-MDP. TS-MDP demonstrates a similar trend as O-TS-MDP since they are both Thompson Sampling-based algorithms. Despite the lack of theoretical analysis, TS-MDP does achieve better empirical performance. The gap between O-TS-MDP and TS-MDP comes from the fact that clipping the left side of the posterior distributions increases the chance to visit a sub-optimal (s,a,t), just as implied in the design of MOTS [Jin et al., 2021]. It is not surprising that SSR-Bernstein outperforms the remaining algorithms as it is theoretically optimal.[1] Similar trends exist for Thomson-sampling-based methods in larger state space, but SSR-Bernstein still performs the best.

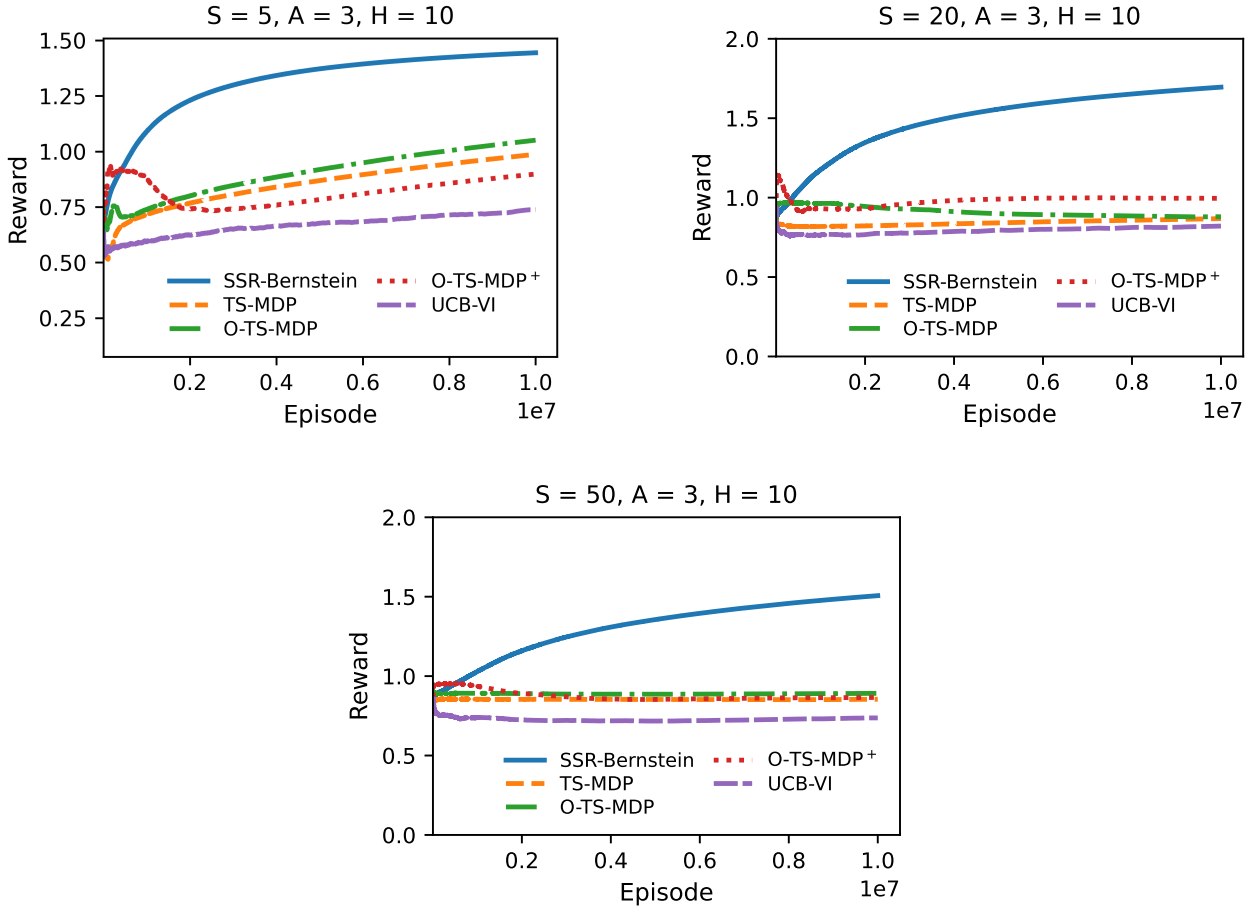

Figure 5.2: Empirical performance for 5, 20 and 50 states, and UCB-VI

It is important to note that SSR-Bernstein uses a single random seed for all (s,a,t) in each episode. In contrast, in our proposed algorithms, each (s,a,t) has its own randomness within an episode. In other words, our proposed algorithms inject more randomness than SSR-Bernstein. Additionally, SSR-Bernstein needs

[1] SSR-Bernstein [Xiong et al., 2022] is a concurrent work published in 2022.

to construct confidence intervals to tune the magnitude of the variance, whereas our algorithms are simpler and easier to implement and enjoy good regret bounds.

We also implement UCB-VI [Azar et al., 2017]. From Figure 5.2, we can see that SSR-Bernstein still performs the best. All three Thompson Sampling-like algorithms perform similarly and better than UCB-VI. All experimental results share similar trends across random initialization.

5.4 Discussion

Both of our algorithms O-TS-MDP and O-TS-MDP$^+$ are based on the idea of modifying the Gaussian distribution in Thompson sampling to be more optimistic. In both algorithms, we encourage exploration by carefully choosing the variance of the distribution and making sure the sampled parameter is optimistic to some extent. O-TS-MDP is both computationally efficient and space efficient, as it improved on existing work and only draws one sample per round for each (s,a,t) pair. The one-sided distribution helps to avoid upper bounding the absolute value of the estimation error and therefore simplifies the theoretical analysis. O-TS-MDP$^+$ is a more optimistic clipping of the posterior distribution and therefore enjoys (near)-optimal regret bound. In the next Chapter, we will discuss future directions and practical modifications of the proposed algorithms.

Chapter 6

Conclusion and Future Work

6.1 Conclusion and Limitations

Balancing between exploration and exploitation is a fundamental challenge in both multi-armed bandit (MAB) problems and reinforcement learning (RL). Exploration involves gathering new information about uncertain actions to make better decisions in the long run, while exploitation involves maximizing the immediate reward based on the current knowledge. Striking the right balance is crucial: too much exploration might lead to unnecessary resource wastage, while too much exploitation could result in missing out on potentially more rewarding actions. Various strategies, such as epsilon-greedy, Thompson sampling (TS) [Agrawal and Goyal, 2017], and upper confidence bound (UCB) [Auer et al., 2002a], have been devised to tackle this dilemma. These methods intelligently allocate resources to explore new actions and exploit known good actions, adapting their behaviour over time to optimize the trade-off between learning and performance, ultimately driving effective decision-making and learning in dynamic environments.

In this work, we discuss various types of regret and optimality in theoretical analysis. We then draw inspiration from TS and UCB and propose the optimistic Thompson sampling (O-TS) algorithm for both bandits and RL. Chapelle and Li [2011] have demonstrated the empirical performance of O-TS for stochastic bandits. We derive regret analysis to fill in the gap theory. In addition, we propose O-TS-Bandit$^+$, an optimism-based learning algorithm, for stochastic bandits. Both O-TS and O-TS-Bandit$^+$ achieve the (order)-optimal problem-dependent regret bound. We demonstrate the performance of our algorithms with a simple numerical experiment.

Next, we extend the ideas of optimistic Thompson sampling to MDPs, a common framework for RL. We first consider various assumptions of MDPs that appear frequently in related works and justify our choices of assumptions. Then, we propose O-TS-MDP, a computationally efficient and theoretically elegant model-based learning algorithm with randomized value functions for episodic MDPs. O-TS-MDP uses a modified Gaussian distribution to model the reward distributions and drive exploration. The clipping of the left side of the posterior distribution simplifies its theoretical analysis. We also propose O-TS-MDP$^+$, a model-based, optimistic algorithm with randomized value functions. O-TS-MDP$^+$ achieves the (near)-

optimal regret bound with a more aggressive clipping strategy of the posterior distribution. It boosts the value of the random sample to the upper confidence bound if the random sample is smaller than the upper confidence bound. The aggressive clipping contributes to reducing the variance of the reshaped posterior distribution as compared to O-TS-MDP. We evaluated our algorithms in a randomized MDP experiment and compared them to the existing methods.

There are some limitations to our work. Firstly, both our theoretical analysis and experimental results are restricted to the tabular setting, with discrete states and actions and a finite time horizon. The tabular assumptions offer simplicity for the theoretical analysis, but they are constrained by their scalability. Storing and updating the values for each state-action pair becomes computationally expensive and memory-intensive when faced with large state and action spaces. In addition, our work is mostly concerned with sample complexity, as backward planning in finite tabular settings is affordable. However, computational efficiency in planning becomes crucial going beyond small MDPs. In the next session, we propose a few future directions for addressing these limitations.

6.2 Future Work

This work opens up many interesting questions and future directions:

Practical Approaches Extending our exploration algorithms to practical applications faces several challenges. One major challenge is the computational efficiency and the memory storage especially when dealing with large state and action spaces. As the number of entries in the Q-table grows exponentially, we can't afford to keep track of the entire history, the exact observations and all possible state transitions. Some approximation methods are needed to make our methods scalable. Another important practical consideration is the robustness of the misspecification of the parameterized value function. Algorithms need to perform well even when the underlying model of the environment is inaccurate, such as when function approximations like neural networks are used.

Many existing approaches aim to extend the idea of exploration via randomization in practice. Bootstrap DQN [Osband et al., 2016a] uses randomized value functions to carry out deep exploration and allows algorithms to learn efficiently. Ensemble sampling [Lu and Roy, 2023] uniformly samples a model from a set of models, acts greedily according to that model, and updates all models to approximate Thompson sampling. Ash et al. [2022] extends UCB-style exploration to deep learning by taking a maximum of a few samples. The idea of random dropout or initialization can also be used for exploration [Burda et al., 2018]. More recently, Mei et al. [2023] showed that noise in stochastic gradient descent can be used for exploration in bandits.

It is interesting to study how our optimistic Thompson sampling style of methods can be approximated and used for exploration in large-scale RL. As a direct extension of our work, one can perhaps use random samples for the transition dynamics in addition to the rewards. With careful analysis and adjustment, noise in optimization and planning procedures may be helpful in exploration as well. Exploration in distributional

RL such as directly injecting noise into policies for policy iteration could also be examined.

Greedy Q updates Most of our work concerns sample efficiency, but it is also important to optimize the computational efficiency in the planning step, especially in practice. The time complexity per episode of our algorithms and many other classical model-based RL algorithms is $O\left(AS^2H\right)$, but it is possible to have fast algorithms. Inspired by Efroni et al. [2019], we propose a rule of greedy updates that can potentially speed up the planning phase in our exploration algorithm.

Algorithm 6.1 Fast-TS

Input: MDP instance M, number of episodes T
Initialization:
Set $n^0_{s,a,h} \leftarrow 0$, $\hat{P}^0_{s,a,h} \leftarrow \vec{0}$, $\hat{R}^0_{s,a,h} \leftarrow 0, \forall (s,a,h)$ % Empirical estimates
Set $\tilde{V}^0_h(s) \leftarrow H-(h-1), \forall (s,h)$ % Global value functions; initialized to the support
for episode $k = 1,2,\ldots,T$ **do**
  Draw $s^k_1 \sim p_0$ % Draw the initial state in episode k from distribution p_0
  for $h = 1,2,\ldots,H$ **do**
    for $a \in \mathcal{A}$ **do**
      Draw $\tilde{R}^k_{s^k_h,a,h} \sim \mathcal{N}_{\text{clipped}}\left(\hat{R}^{k-1}_{s^k_h,a,h}, \widetilde{O}\left(\frac{SH(H-h+1)^2}{n^{k-1}_{s^k_h,a,h}}\right)\right)$
      Compute $\tilde{Q}^k_{s^k_h,a,h} = \tilde{R}^k_{s^k_h,a,h} + \left\langle \hat{P}^{k-1}_{s^k_h,a,h}, \tilde{V}^{k-1}_{h+1} \right\rangle$ % Only compute Q values of all the actions for the visited state
    end for
    Find $a^k_h \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{Q}^k_{s^k_h,a,h}$
    Update $\tilde{V}^k_h(s^k_h) \leftarrow \min\left\{\tilde{V}^{k-1}_h(s^k_h), \tilde{Q}^k_{s^k_h,a^k_h,h}\right\}$
    Play a^k_h and transition to s^k_{h+1}
  end for
  Observe a trajectory $\left(s^k_1,a^k_1,r^k_{s^k_1,a^k_1,1}\right), \left(s^k_2,a^k_2,r^k_{s^k_2,a^k_2,2}\right), \ldots, \left(s^k_h,a^k_h,r^k_{s^k_h,a^k_h,h}\right), \ldots, \left(s^k_H,a^k_H,r^k_{s^k_H,a^k_H,H}\right)$
  Update the empirical estimates of the visited states and actions.
end for

In our fast algorithm, the time complexity per episode is $O(ASH)$ instead of $O(AS^2H)$. We use a global value function to keep track of previous estimates instead of computing this estimation for each state-action-time pair at every round. This algorithm avoids computing the Q values of all the states for all the actions in each round. We only compute the Q values for all the actions associated with the visited trajectory. Preliminary experimental results confirm this speedup, but further analysis is needed to evaluate whether the convergence to optimality is sacrificed. We suspect that by adding less noise to the model, the practical performance might be improved. Then, the next step would be a possible extension to the infinite horizon.

Connection to DP Differential privacy (DP) [Dwork et al., 2014] is a framework for enhancing the privacy of individuals in data analysis by introducing controlled noise to the results while still providing meaningful

insights. It aims to prevent the disclosure of specific information about any individual's data in a dataset. Privacy is crucial in machine learning to safeguard sensitive information, prevent unintended disclosure of personal data, and ensure ethical and secure deployment of AI agents in real-world environments.

One common technique within differential privacy is the Gaussian mechanism [Dong et al., 2019], which involves adding noise sampled from a Gaussian (normal) distribution to the output of a computation. The amount of noise added is calibrated to achieve a balance between privacy and accuracy. The Gaussian mechanism is widely used due to its mathematical properties and ease of implementation, offering a practical way to achieve differential privacy in various data analysis and machine learning tasks, thereby safeguarding individual privacy while permitting useful data analysis.

There are a lot of connections between the Gaussian noises added for differential privacy, and the Gaussian noises added for exploration in bandits and RL. The key ingredient is the variance of the noise. In DP, too much noise would lead to loss of information, but too little noise might leak private information. In RL, too much noise would cause divergence from the optimal policy, but too little noise prevents exploration and therefore might be stuck at sub-optimal solutions. We believe further studies of this connection could provide insights into both fields. It is our ongoing project to design practical exploration methods by perturbing or injecting noise into the training data and at the same time guarantee privacy.

Bibliography

Alekh Agarwal, Nan Jiang, and Sham M. Kakade. Reinforcement learning: Theory and algorithms. 2019. URL https://api.semanticscholar.org/CorpusID:148567317. → pages 31, 32

Priyank Agrawal, Jinglin Chen, and Nan Jiang. Improved worst-case regret bounds for randomized least-squares value iteration. *ArXiv*, abs/2010.12163, 2020. URL https://api.semanticscholar.org/CorpusID:225062284. → pages 35, 40

Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *J. ACM*, 64(5), sep 2017. ISSN 0004-5411. doi:10.1145/3088510. URL https://doi.org/10.1145/3088510. → pages v, 3, 12, 18, 22, 24, 35, 37, 45

Jordan T. Ash, Cyril Zhang, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Anti-concentrated confidence bonuses for scalable exploration, 2022. → page 46

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. *Annual Symposium on Foundations of Computer Science - Proceedings*, pages 322–331, December 1995. ISSN 0272-5428. Proceedings of the 1995 IEEE 36th Annual Symposium on Foundations of Computer Science ; Conference date: 23-10-1995 Through 25-10-1995. → page 9

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002a. → pages v, 3, 12, 14, 16, 24, 45

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b. → page 10

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning, 2017. → pages 4, 34, 41, 44

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966. → page 29

Djallel Bouneffouf and Irina Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*, 2019. → page 5

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. → page 46

Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3690147. → page 9

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. → pages iii, v, 3, 12, 19, 24, 35, 38, 45

Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109: 101964, 2020. → pages 1, 26

Christoph Dann, Tor Lattimore, and Emma Brunskill. UBEV - A more practical algorithm for episodic RL with near-optimal PAC and regret guarantees. *CoRR*, abs/1703.07710, 2017. URL http://arxiv.org/abs/1703.07710. → pages 10, 34, 37, 42

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019. → page 48

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. → page 47

Yonathan Efroni, Nadav Merlis, Mohammad Ghavamzadeh, and Shie Mannor. Tight regret bounds for model-based reinforcement learning with greedy policies. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. → page 47

J. C. Gittins and D. M. Jones. A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika*, 66(3):561–565, 1979. ISSN 00063444. URL http://www.jstor.org/stable/2335176. → page 10

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017. → pages 1, 26

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994. → page 15

Bingshan Hu and Nidhi Hegde. Near-optimal thompson sampling-based algorithms for differentially private stochastic bandits. In *Uncertainty in Artificial Intelligence*, pages 844–852. PMLR, 2022. → page 37

Bingshan Hu, Tianyue H. Zhang, Nidhi Hegde, and Mark Schmidt. Optimistic Thompson sampling-based algorithms for episodic reinforcement learning. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 890–899. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/hu23a.html. → pages v, 19, 22, 27, 37, 38

Tianyuan Jin, Pan Xu, Jieming Shi, Xiaokui Xiao, and Quanquan Gu. Mots: Minimax optimal thompson sampling. In *International Conference on Machine Learning*, pages 5074–5083. PMLR, 2021. → pages 37, 43

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite time analysis, 2012. → page 35

T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. → page 9

Tor Lattimore and Csaba Szepesvri. *Bandit Algorithms*. Number 10.1017/9781108571401. Cambridge University Press, 2020. → page 5

Yang Li, Wanshan Zheng, and Zibin Zheng. Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access*, 7:108014–108022, 2019. → page 1

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling, 2023. → page 46

Jincheng Mei, Zixin Zhong, Bo Dai, Alekh Agarwal, Csaba Szepesvari, and Dale Schuurmans. Stochastic gradient succeeds for bandits. 2023. → page 46

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. → page 32

Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning?, 2017. → pages 34, 36

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *CoRR*, abs/1602.04621, 2016a. URL http://arxiv.org/abs/1602.04621. → page 46

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions, 2016b. → page 36

Ian Osband, Benjamin Van Roy, Daniel Russo, and Zheng Wen. Deep exploration via randomized value functions, 2019. → pages 34, 35

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. → page 1

Aldo Pacchiano, Philip Ball, Jack Parker-Holder, Krzysztof Choromanski, and Stephen Roberts. Towards tractable optimism in model-based reinforcement learning. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1413–1423. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/pacchiano21a.html. → pages 35, 41

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018. → page 1

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. → page 31

Herbert Robbins. Some aspects of the sequential design of experiments. 1952. → pages 2, 5

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions, 2019. → pages 37, 40

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*, 2017. → page 1

Eric M Schwartz, Eric T Bradlow, and Peter S Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017. → page 5

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016. → pages 1, 26

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014. → page 32

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. → pages 1, 26

Farzan Soleymani and Eric Paquet. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—deepbreath. *Expert Systems with Applications*, 156: 113456, 2020. → pages 1, 26

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html. → page 12

William 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. → page 17

Daniil Tiapkin, Denis Belomestny, Eric Moulines, Alexey Naumov, Sergey Samsonov, Yunhao Tang, Michal Valko, and Pierre Menard. From Dirichlet to rubin: Optimistic exploration in RL without bonuses. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21380–21431. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/tiapkin22a.html. → pages 34, 40, 41

Michel Tokic. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, pages 203–210. Springer, 2010. → page 13

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015. → page 5

Lilian Weng. A (long) peek into reinforcement learning. *lilianweng.github.io*, 2018. URL https://lilianweng.github.io/posts/2018-02-19-rl-overview/. → pages ix, 26

Zhihan Xiong, Ruoqi Shen, Qiwen Cui, Maryam Fazel, and Simon S. Du. Near-optimal randomized exploration for tabular markov decision processes, 2022. → pages 4, 35, 40, 41, 42, 43

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011. → page 33

Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021. → page 1

Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learning via reference-advantage decomposition, 2020. → page 34

Kai Zhu and Tao Zhang. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*, 26(5):674–691, 2021. → page 1

□