To Ask or Explore: A Systematic Approach to Advice

by

Amirmohsen Sattarifard

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Reinforcement learning (RL) has shown great promise in sequential decision-making tasks. However, one of the significant challenges RL faces is poor sample efficiency, which restricts its applicability in many real-world scenarios. Addressing this challenge has the potential to expand the reach of RL techniques. One class of problems that offer insights into this challenge is the multi-armed bandits (MAB) setting, which can be viewed as a simplified version of RL. By investigating and addressing sample efficiency in MAB, we hope to gain insights that can later be generalized to broader RL contexts.

In this thesis, our primary focus is the Beta-Bernoulli Bayesian multi-armed bandit with an online finite horizon setting. We adopt two strategies to tackle the sample efficiency challenge: 1- Knowledge reuse through advice and 2- Near-optimal exploration. While the advice-seeking approach has been touched upon in earlier research, it has largely been unstructured. Our work aims to provide a more systematic approach to the problem of advice in MAB, addressing two essential questions: "when to ask for advice?" and "what arm to ask about?". Finally, we investigate the problem of near-optimal exploration. We provide a myopic approximation to the Bayes-optimal policy and show that we can reach near-optimal performance using a myopic approximation.

Our key contributions include:

- Assuming an advice budget of one, we derive an optimal solution for

which arm the agent should seek advice, using the value of information (VOI) as a metric for each advice's utility.

- Providing and investigating some approximations of the VOI based on our myopic horizon approximation to the Bayes-optimal policy.
- Drawing parallels between the Bayes-optimal policy and the value of information, and revealing the intertwined nature of these two aspects.
- Demonstrating that in some cases, it is beneficial for an agent to postpone seeking advice, even when it is free, and providing the optimal solution for such delays in a Bayesian bandit setting.
- Investigating a myopic horizon approximation to the Bayes-optimal policy and showing its efficacy in achieving near-optimal results.

To my mother

whose unwavering support and encouragement have been the driving force behind my academic journey.

The greatest glory in living lies not in never falling,
but in rising every time we fall.

– Nelson Mandela

Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have contributed to the completion of this master's thesis. Their support, guidance, and encouragement have been invaluable throughout this challenging journey.

First and foremost, I am immensely grateful to my supervisors, James Wright and Matt Taylor, for their unwavering guidance, bearing with me through ups and downs, and constant encouragement. Their insightful feedback, constructive criticism, and commitment to excellence have played a pivotal role in shaping the direction and quality of this research. I am truly fortunate to have had their mentorship, which has not only enhanced my academic growth but also enriched my overall learning experience.

I am indebted to my mother and sister for their unwavering support and understanding throughout this journey. Their love, encouragement, and belief in my abilities have been the driving force behind my determination to complete this thesis. Their constant presence and encouragement, even during the most challenging times, have meant the world to me.

I am also grateful to my friends, Kiarash, Azali and Pooyayi, Nazi and Farazi, Bedir, Hamza, and many others. They have supported me throughout this process. Their words of encouragement have provided much-needed solace during moments of stress.

Finally, I want to thank me for doing all this hard work. I want to thank me for never quitting. I want to thank me for always being a giver and trying to give more than I receive. I want to thank me for trying to do more right and wrong. I want to thank me for just being me at all times.

In conclusion, this master's thesis represents the culmination of countless

hours of hard work, support, and guidance from numerous individuals. Each person mentioned, and many others who may not be explicitly named have contributed in their own unique way to the successful completion of this research. I am truly grateful for their contributions and the lasting impact they have had on my academic and personal growth.

Thank you all from the bottom of my heart.

Amir

Contents

List of Figures

List of Symbols

The notations used throughout this thesis are summarized in the table below.

Note: these notations are defined in the order they are first encountered in the main body of this thesis.

Symbol	Description
k	Number of arms in a multi-armed bandit (k-armed bandit).
$T \in \mathbb{N}^+$	The problem setting's horizon.
$t \in [T]$	Time step.
$n \in [T]$	number of time steps left till the end of the horizon: $n = T - t + 1$
$\mathcal{A} = \{1, 2, \ldots, k\}$	Set of possible actions.
$\mathcal{R}$	The set of all possible reward outcomes.
$\mathcal{P}(\mathcal{R})$	The set of all possible probability density functions of rewards for all the arms.
$R \in \mathcal{P}(\mathcal{R})$	The probability density function (PDF) of the reward for all the arms.
$R(\cdot\|a)$	The PDF of rewards for arm a.
$a_t \sim A_t$	Agent's action at time step t, and the random variable associated with agent's action at time step t, respectively.
$X_t^a \in \mathfrak{R}$	The random variable associated with the PDF of arm a at time step t
a^*	Represents the optimal single action to take: $\arg\max_a \mathbb{E}[X^a]$
θ	Represents the parameters of the environment (e.g. bandit model)
$\mathcal{B}$	Denotes the set of possible belief distributions that the agent can hold over the parameters θ of the environment. $b_0 \in \mathcal{B}$ is the agent's prior belief over the parameters of the environment.
$b_t \in \mathcal{B}$	The agent's belief about the parameters of the environment at time step t. b_t^i is the agent's belief at time step t about arm i.
$\mathcal{S}$	The set of states (state space)
A_s	Represents the set of actions available at state s
T	The probability density function of the transition for an MDP, which conditions the next state on the current state and action: $T(s'\|s, a) == Pr(s_{t+1} = s'\|s_t = s, a_t = a)$

Symbol	Description
p	The probability p is the parameter of the Bernoulli distribution. In a single Bernoulli trial, p represents the probability of the event of interest (usually termed "success") occurring.
(α, β)	These are the parameters of the Beta distribution, which in the case of the Beta-Bernoulli model, represent the prior knowledge about the probability of success (usually denoted as p) in a Bernoulli trial. Alpha (α): This parameter can represent the number of prior successes. Beta (β): This parameter represents the number of prior failures.
$Beta(\alpha, \beta)$	The Beta function, denoted as $Beta(\alpha, \beta)$, is defined for positive α and β as: $Beta(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$. The Beta function is a special mathematical function that arises in probability theory, particularly in the normalization of the Beta distribution.
$b_{\|x}$	Represents $b(\theta \mid x)$ for the simplicity.
R	The probability density function of the reward for an MDP, which conditions the immediate reward on the current state and action: $R(r\|s, a) == Pr(r_{t+1} = r\|s_t = s, a_t = a)$
$\mathcal{Z}$	The set of possible observations (observation space).
$z_t \sim Z_t$	observation at time step t, and the random variable associated with the observation at time step t.
O	The probability density function of the observation for a POMDP, which conditions the current observation on the current state and the last action taken. It can be defined as follows: $O(s, a, z) = Pr[z_t = z\|s_t = s, a_{t-1} = a]$
V	The value function, which can be defined as $V = \mathbb{E}[\sum_{t=0}^{n} r_t]$
$u_i(b, a, o, \Theta)$	The update function. After observing an observation z, the agent updates its belief using the update function defined as $u_i(b, a, z, \Theta) = Pr[s_{t+1} = i \mid a_t = a, b_t = b, z_{t+1} = z, \theta]$.
$\mathcal{M}$	The set of all possible MDPs
μ_a	The mean reward for arm or action a: $\mu_a = \mathbb{E}[r_a]$
Q_a	Represents the unbiased estimate of the mean reward of arm a (μ_a)
$\mathcal{M}$	The set of all possible MDPs for a specific function class valid for the environment
$\Omega = \mathcal{S} \times \mathcal{B}$	The hyper-state in a BAMDP, where $\mathcal{B}$ is the set of all possible beliefs (probability measures) on $\mathcal{M}$.

Symbol	Description
T'	The probability density function of the BAMDP's transition, which conditions the next state and belief on the current state, belief, and action. It can be defined as $T'(s', b'\|s, b, a) = \Pr(s_{t+1} = s', b_{t+1} = b'\|s_t = s, b_t = b, a_t = a)$
R'	The probability density function of the reward for the BAMDP, which conditions the immediate reward on the current state, belief, and action. It can be defined as $R'(r\|s, a) = \Pr(r_{t+1} = r\|s_t = s, b_t = b, a_t = a)$
Φ	Represent the advice budget.

Chapter 1

Introduction

The field of reinforcement learning (RL) [20] has seen its applications expand to tackle increasingly intricate problems, such as robotics tasks. While RL remains one of the most effective methods for autonomous learning in sequential decision-making problems, RL needs interaction with the environment to gather samples. This interaction can make the learning process prohibitively expensive or hazardous for various real-world applications. As a result, recent investigations focus on minimizing the number of required samples during the learning process. Sample efficiency in RL can be accomplished by incorporating external information or enhancing the RL algorithms.

This thesis dives deep into discovering efficient strategies for incorporating external information and enhancing the RL algorithms to bridge this gap and enhance the efficiency and feasibility of RL. For ease of computation, this thesis shifts the focus to the multi-armed bandits (MAB) setting, a simplified version and a critical component of RL. MAB provides a more controlled and manageable environment, allowing for more detailed and focused exploration strategies to improve sample efficiency. The selection of the MAB setting offers a platform to address the significant challenge of sample efficiency, reinforcing the interdependence between RL and bandit problems and underscoring the importance of MAB. In this thesis, we specifically use the Beta-Bernoulli Bayesian multi-armed bandit with a finite horizon since Bernoulli distribution only has two outcomes, and helps to make the analysis computationally tractable.

In the case of the first solution, incorporating external information, a learning agent (student) can leverage information (referred to as the "advice") provided by an oracle that possesses knowledge of the true underlying parameter of the environment. This advice helps minimize the need for random exploration, which can be potentially risky or harmful. However, the use of advice presents some challenges. For example, the availability of advice is often constrained by human availability, or there is a communication cost. Consequently, the learning agent must optimize the use of the available advice. This research examines what is the optimal time to ask for advice and which arm will provide the most useful advice, employing the metric of the value of information (VOI) to evaluate the utilities of each piece of advice. By providing a detailed analysis and myopic approximations to the Bayes-optimal policy, this work illuminates the relationship between VOI and the Bayes-optimal strategy.

As for the second solution, enhancing the sample efficiency of RL algorithms, optimal approaches such as the Bayes-optimal policy cannot be employed due to their computational intractability. Therefore, it becomes necessary to devise approximations that are both sufficiently close to the optimal solution and computationally feasible. We achieve this approximation by using myopic estimation of the environment's underlying horizon.

The following chapters offer are organized as follows. Chapter 2 lays down the background foundation, illuminating the underlying concepts of Bayesian learning, Bayes-optimal policy, and VOI. Chapters 3 and 4 present a structured exploration of the advice-seeking problem, providing clear solutions and approaches for the "When to ask for advice problem"and the "when to ask for advice problem," respectively. Chapter 5 further explores the Bayes-optimal policy and provides insight into its efficacy through myopic approximations, illustrating their potential to achieve near-optimal results. Finally, we conclude with a summary of our findings and suggestions for future research in Chapter 6.

This thesis contributes to the ongoing efforts in understanding and enhancing the sample efficiency in both the multi-armed bandit settings and

the broader RL contexts, thereby promising a pathway towards more efficient, effective, and practical RL applications in diverse real-world scenarios.

Chapter 2

Background Material

This chapter provides an overview of the key concepts and frameworks relevant to this thesis. We begin by describing the multi-armed bandit setting (2.1), which serves as the primary focus of our study. We chose this setting due to its simplicity and its suitability for analyzing the algorithms discussed in the subsequent chapters (3, 4, and 5). After establishing the multi-armed bandit, we dive into Bayesian learning in a specific environment known as the Beta-Bernoulli multi-armed bandit in Section 2.2. We then proceed to explain various formulations of sequential decision-making known as Markov decision processes (MDPs), partially observable Markov decision processes (POMDPs), and belief-augmented MDPs (BAMDPs) in Sections 2.3, 2.4, and 2.4.2, respectively. Next, we discuss VOI (Section 2.5) and its variation in the context of POMDPs. And also, we discuss definitions that are prerequisites for understanding the material presented in Chapters 3, 4, and 5. Finally, in Section 2.6, we explain chosen baseline algorithms, such as the Bayes-optimal exploration strategy and Thompson sampling.

2.1 Stochastic Multi-Armed Bandit

In the fields of machine learning and probability theory, a multi-armed bandit (k-armed bandit) refers to a sequential decision-making process involving a learner (agent) and an environment. The goal is to maximize the expected cumulative rewards [15]. Some of the bandit's parameters are not known to the agent beforehand. In this thesis, the environment is only partially

observable since we want to be as close to conventional RL environments as possible. The interaction between the agent and the environment consists of T rounds, where $T \in \mathbb{N}^+$ represents the horizon. In each round, at time step $t \in [T]$, the agent selects an action (an arm) $a_t \sim A_t$, in which A_t is the random variable associated with the actions at time step t, from a given set of possible actions $\mathcal{A} = \{1, 2, \ldots, k\}$. Subsequently, the agent receives an observation from the environment, typically in the form of rewards $x_t \sim X_t \in \mathfrak{R}$. The observed reward can be a stochastic value that varies based on the environment's underlying reward distributions. The agent only has access to the previous actions and observations, collectively referred to as the history $H_{t-1} = \{A_1, X_1, \ldots, X_{t-1}, A_{t-1}, X_t\}$.

Here, we present a formal model of the stochastic k-armed bandit as a tuple $\langle \mathcal{A}, \mathfrak{R}, \mathcal{P} \rangle$:

- $\mathcal{A}$ represents the set of actions (arms), where $|\mathcal{A}| = k$.
- $\mathcal{R}$ denotes the set of all possible reward outcomes $\mathcal{R}$ for all the arms.
- $R \in \mathcal{P}(\mathcal{R})$ denotes the probability density function (PDF) of the reward for all the arms, in which $\mathcal{P}(R)$ denotes the set of all possible probability density functions of rewards for all the arms.
    - $R(\cdot|a)$ is the PDF of rewards for arm a.
    - X^a is the random variable associated with $R(\cdot|a)$.

In this problem setting, a policy refers to a mapping from past observations (histories) to a distribution over the set of actions, which the agent learns in order to interact with the environment. The goal of this interaction is to plan actions in a manner that maximizes the expected return. There are other performance measures, such as regret, which have different definitions depending on the frequentist or Bayesian perspective.

If the performance measure is the expected return, defined by the formula $\mathbb{E}[\sum_{t=1}^{T} x^{a_t}]$, the objective is to maximize this expected return. Similarly, if the performance measure is frequentist regret, given by the formula

$\mathbb{E}[\sum_{t=1}^{T}(x^{a^*} - x^{a_t})]$ where a^* is the optimal action, $\arg\max_a \mathbb{E}[X^a]$, the objective is to minimize this frequentist regret. The objective of optimizing the agent's policy is based on the expected return, while regret serves as a performance measure. In cases where there is a prior belief about the environment, it is common to use Bayesian regret (Definition 1 in Section 2.4.1); otherwise, frequentist regret is employed. In this work, Bayesian regret will be used since we are in a Bayesian setting and hold a prior belief regarding the environment's parameters.

Due to the agent's lack of knowledge regarding the environment's underlying probability distribution, a trade-off arises between exploration and exploitation at each step. Exploration involves gathering more information about the environment and potentially discovering the most rewarding actions, while exploitation focuses on selecting the most rewarding action as frequently as possible based on the agent's current knowledge, referred to as the belief b. This trade-off is a fundamental challenge known as the exploration-exploitation dilemma.

2.2 Bayesian Learning in Multi-Armed Bandit

This thesis focuses on parametric bandit models due to the ease of incorporating prior knowledge. Parametric models can more readily incorporate prior knowledge through the choice of the distribution and its parameters. This can be especially useful in domains where prior knowledge is available such as Bayesian learning. A parametric bandit model is a type of MAB model where the reward distributions of each arm are parameterized by some known functions of unknown parameters. The task in a parametric bandit problem is to learn the unknown parameters while simultaneously optimizing the cumulative reward or minimizing regret. In learning theory, two main approaches are commonly used: the frequentist approach and the Bayesian approach. The key distinction between these approaches lies in the ability of the Bayesian approach to incorporate probability distributions over the parameters of the bandit model. In the following sections, we explore the fundamentals of Bayesian

stochastic multi-armed bandits and discuss a specific case known as the Beta-Bernoulli multi-armed bandit, which serves as the problem setting for the subsequent chapters (3, 4, and 5).

2.2.1 Bayesian Stochastic Multi-armed Bandit

In this problem-setting, in addition to the tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$ representing the multi-armed bandit, we also maintain a belief distribution over the parameters of the environment. Here is the addition to the model in Section 2.1, defining the Bayesian stochastic k-armed bandit as a tuple $\langle \mathbf{b_0}, \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$:

- $\mathbf{b_0} \in \mathcal{B}$ is the prior belief while $\mathcal{B}$ denotes the set of possible belief distributions that the agent can hold over the parameters of the environment.
  - At each time step, after receiving an observation from the environment, the agent updates its previous belief b_{t-1} to obtain a new belief b_t (posterior distribution).

As mentioned earlier, the probability distribution over the parameters θ of the bandit model is represented as the belief b. The transition (update rule) between different beliefs after each observation is achieved through Bayesian inference. The following subsection dive into the Beta-Bernoulli distribution, a fundamental component of our Beta-Bernoulli bandit framework. followed by a review of the Bayesian update rule.

2.2.2 Bernoulli Distribution with Beta Prior Belief

Bernoulli Distribution

The Bernoulli distribution is a discrete probability distribution that models binary events with a single parameter p. It represents the probability of success, denoted as p, or the probability of observing a reward $x = 1$, while the probability of failure or observing a reward $x = 0$ is $1 - p$. The probability mass function (PMF) of the Bernoulli distribution is given by:

$$f(x \sim X \mid p) = \begin{cases} p & \text{if } x = 1 \\ q = 1 - p & \text{if } x = 0 \end{cases} \tag{2.1}$$

In Equation 2.1, X is the random variable representing the reward, which can take on the values 0 or 1. The mean of the Bernoulli distribution is given by $\mathbb{E}(X) = p$, and the variance is $Var(X) = pq = p(1-p)$.

Beta Distribution

The Beta distribution can be used as a prior belief for the Bernoulli distribution. The Beta distribution is a continuous probability distribution defined on the interval $[0, 1]$. This distribution is widely employed in learning theory as it serves as a conjugate prior distribution for the binomial and Bernoulli distributions. The Beta distribution has two parameters, denoted as α and β, both of which are positive. The probability density function (PDF) of the Beta distribution is given by:

$$B(\alpha, \beta) \stackrel{\Delta}{=} f(x \mid \alpha, \beta) = \frac{1}{Beta(\alpha, \beta)} x^{(\alpha-1)}(1-x)^{(\beta-1)} \tag{2.2}$$

$$Beta(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt \tag{2.3}$$

In Equation 2.2, $Beta(\alpha, \beta)$ (Equation 2.3 is the Beta function, which acts as a normalizer for the PDF to be a valid probability distribution, which is to ensure that the total area under the PDF curve equals 1. The mean of the Beta distribution is given by $\mathbb{E}(X) = \frac{\alpha}{\alpha+\beta}$, and the variance is $Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$. The variance of the Beta distribution decreases as α and β become larger, resulting in a distribution that is more concentrated around the mean. Notably, the Beta distribution with parameters $\alpha = 1$ and $\beta = 1$ is equivalent to a uniform distribution.

Modeling the Bernoulli Parameter With Beta Distribution

When modeling the parameter p of a Bernoulli distribution, the most common choice is to choose conjugate priors such as the Beta distribution in this thesis. The reason behind this choice is that the posterior distribution is also a Beta distribution, which simplifies the computations of calculating the posterior distribution. We consider $\theta = p$ as the parameter of interest in the Bernoulli

distribution, and we use the Beta distribution as the probability distribution for modeling the Bernoulli parameter. The probability density function (PDF) of the Bernoulli parameter model given the parameters α and β of the Beta distribution is as follows.

$$f(p \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} p^{(\alpha-1)}(1-p)^{(\beta-1)} \tag{2.4}$$

2.2.3 Bayesian Update

Bayesian statistics employs a framework where beliefs about the unknown parameters of the environment (in this case, the multi-armed bandit) are represented by probability distributions [12]. Let θ denote the parameter of arm a of a multi-armed bandit model, in which $P(\cdot|a) = P(\cdot|\theta_a)$ is the reward distribution for arm a with parameter θ. Also, let $b_t = Pr(\theta \mid x)$ represent the posterior density that reflects our belief about the value of θ after observing x. Furthermore, let $b_0 = Pr(\theta)$ be our prior belief about the parameter of the multi-armed bandit model. The update rule for transforming the prior distribution into the posterior distribution after an observation y is given by Bayes' rule:

$$b_1(\theta) = b_0(\theta \mid x) = \frac{Pr(x \mid \theta) b_0(\theta)}{\int_\theta Pr(x \mid \theta) b_0(\theta)}$$

This update rule can be extended to a sequence of observation steps, making it suitable for sequential decision-making problems [8]. The posterior belief at each step becomes the prior for the next step, as expressed by the equation:

$$b_t(\theta) = b_{t-1}(\theta \mid x) = \frac{Pr(x \mid \theta) b_{t-1}(\theta)}{\int_\theta Pr(x \mid \theta) b_{t-1}(\theta)} \tag{2.5}$$

For convenience, we denote $b(\theta \mid x)$ as $b_{|x}$ from now on.

Although this framework is very general, it is useful for the families of beliefs called conjugates, in which the prior and the posterior have the same distribution family. One of these distribution families is called Beta distribution (refer to Section 2.2.2 for more information about Beta distribution). For Bernoulli observations, the Beta family provides conjugate beliefs, which gives us a closed-form solution for the update rule.

Consider a Bernoulli multi-armed bandit with k arms, where $\theta_i = p_i$ represents the parameter for the i_{th} arm, and the parameter model follows a Beta distribution with parameters α and β as the belief $b_t = B(\alpha_t, \beta_t)$. The update rule for the i_{th} arm, based on Equation 2.5, is as follows:

$$b_t^i = B(\alpha_t^i, \beta_t^i) \leftarrow b_{t-1|X_t}^i = B(\alpha_{t-1}^i + X_t^i, \beta_{t-1}^i + (1 - X_t^i)) \tag{2.6}$$

Here, X_t^i represents the observation reward variable for arm i at time-step t, and the tuple (α_t^i, β_t^i) represents the parameters for the Beta distribution of arm i at time-step t. From now on, whenever we talk about belief b for a Beta-Bernoulli environment, we only use the Beta parameters $b = \begin{bmatrix} (\alpha_1, \beta_1) \\ \vdots \\ (\alpha_k, \beta_k) \end{bmatrix}$ to denote the Beta-Bernoulli beliefs, which is equivalent to $b = \begin{bmatrix} B(\alpha_1, \beta_1) \\ \vdots \\ B(\alpha_k, \beta_k) \end{bmatrix}$.

Example 2.2.1. *Consider an example of a Beta-Bernoulli multi-armed bandit with two arms and a uniform distribution ($\alpha = 1, \beta = 1$) as the prior belief for each arm. The prior belief for time 0 is thus given by $b_0 = \begin{bmatrix} (1,1) \\ (1,1) \\ (1,1) \end{bmatrix}$. Now, suppose the agent chooses arm 2 as the action to take and on time step 1 receives a reward of $X_1^2 = 1$. Based on Equation 2.6, the agent updates its belief to $b_1 = b_{0|X_1^2=1} = \begin{bmatrix} (1,1) \\ (2,1) \\ (1,1) \end{bmatrix}$.*

2.3 Markov Decision Process

A Markov decision process (MDP) is a type of stochastic sequential decision process. It provides a framework for modeling the dynamics of an environment and solving optimization problems such as reinforcement learning or the multi-armed bandit setting. In an MDP, decisions are made at discrete time steps, and the agent's actions are determined solely based on the information available from the current state.

At each time step t, the agent is in an observed state s_t and selects an action a_t. This decision leads to the environment transitioning to the next state

s_{t+1} according to its underlying dynamic function T, which is conditionally independent of the history. Additionally, the agent receives a corresponding reward x_t based on the reward function R.

A Markov decision process is defined by the collection of the tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where:

- $\mathcal{S}$ is the set of states (state space).
- $\mathcal{A}$ is the set of actions (action space), where A_s represents the set of actions available at state s.
- $T \in \mathcal{T}$ is the probability density function of the transition for an MDP, which conditions the next state on the current state and action.

  $$T(s'|s,a) = Pr(s_{t+1} = s'|s_t = s, a_t = a)$$

  The transition PDF satisfies the Markov property:

  $$Pr(s_{t+1}|s_t, a_t) = Pr(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots)$$

- R is the probability density function of the reward for an MDP, which conditions the immediate reward on the current state and action.

  $$R(x|s,a) = Pr(x_{t+1} = x|s_t = s, a_t = a)$$

The state and action spaces can be either finite or infinite, but in this thesis, we assume they are finite. The rewards are real numbers, and in the bandit setting, their distributions' parameters are unknown to the agent.

The objective of the agent in an MDP is to find a policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, which represents a distribution over actions, that maximizes the performance over the decision-making horizon. In this thesis, we assume that a policy is deterministic $\pi : \mathcal{S} \to \mathcal{A}$, which is a mapping from a state to an action. The policy aims to optimize the agent's return, which is the sum of rewards, over time. To achieve this, the agent needs to consider the future consequences of its actions. More specifically, the agent seeks a policy π that maximizes the sum (or discounted sum with a discount factor γ) of rewards over the

horizon, not just the immediate step. It is worth noting that the k-armed bandit setting mentioned earlier in Section 2.1 is a special case of an MDP with only one state.

2.4 Partially Observable Markov Decision Process

The partially observable Markov decision process (POMDP) framework is similar to the MDP framework but it is more general. In an MDP, the environment's state $s_t \in \mathcal{S}$ (a finite set) is observed by the agent, which selects an action $a_t \in \mathcal{A}$. Based on the current state s_t and the action taken a_t, the agent receives a reward x_t. The expected reward is determined by the reward function $R(s, a) = E[x_t | s_t = s, a_t = a]$. Following the agent's action, the state changes, possibly stochastically, based on the transition function $T(s, a, s') = P[s_{t+1} = s' | s_t = s, a_t = a]$.

Unlike in an MDP, the POMDP does not assume full observability of the state s_t. Instead, at each time step, the agent only has access to partial information about the current state, summarized by the observation $z_t \in \mathcal{Z}$. The observation function $O(s, a, z) = P[z_t = z | s_t = s, a_{t-1} = a]$ defines the relationship between observations and states.

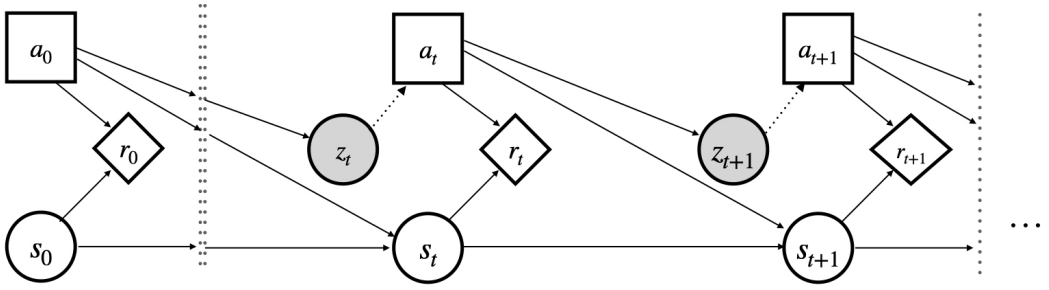

Figure 2.1: Decision graph for a POMDP. Squares represent decision variables, circles represent random variables, diamonds represent utility variables, and arrows indicate dependencies among the mentioned variables. The double-dotted line represents the hidden steps $\{1, \ldots, t-1\}$.

Figure 2.1 presents a graphical model representation of a POMDP, as de-

scribed in Barber's textbook [2]. As mentioned earlier, the agent has partial observations (depicted by the grey variables). At time step t_0, the hidden state is s_0, and the agent selects action a_0, receiving reward x_0. Subsequently, time steps pass, leading the state to transition to s_1, but the agent can only observe a partial observation z_1. After observing z_1, the agent selects action a_1 based on the observation, receives reward x_1, and the state transitions to s_2.

In this problem setting, in addition to the tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$ representing the MDP, we also maintain a belief distribution $\mathbf{b_0}$ over the parameters of the environment, observation space $\mathcal{Z}$, and observation probability distribution. Here is the formal model of the POMDP as a tuple $\langle \mathcal{S}, \mathcal{Z}, \mathcal{A}, \Theta = \{T, O, R\}, \mathbf{b}_0 \rangle$:

- $\mathcal{Z}$ is the set of possible observations (observation space).
- O The probability density function of the observation for a POMDP, which conditions the current observation on the current state and the last action taken. It can be defined as follows:

$$O(z \mid a, s) = Pr[z_t = z | s_t = s, a_{t-1} = a]$$

- $\mathbf{b}_0$ is the agent's initial belief, which represents the agent's prior distribution over states. The agent's belief at time step t based on the history $\{\bar{a}_t, \bar{z}_t\}$ is denoted as b_t and can be defined as follows:
  - $\bar{a}_t = \{a_0, \ldots, a_{t-1}\}$ is the history of actions.
  - $\bar{z}_t = \{z_1, \ldots z_t\}$ is the history of observations.

The POMDP framework allows for modeling decision-making problems in which the agent has incomplete information about the underlying state of the environment. By maintaining a belief state and incorporating observations, the agent can make informed decisions based on the available information, aiming to maximize its expected cumulative reward over time.

2.4.1 The Bayes-Optimal Policy in POMDPs

This section discusses the Bayes-optimal policy in the context of finite horizon partially observable Markov decision processes (POMDPs). The objective of the agent is to maximize the value function V, which represents the expected cumulative rewards over a finite horizon. The value function can be defined as $V = \mathbb{E}[\sum_{t=0}^{n} r_t]$.

At time step t, the agent's knowledge about the current state is represented by a probability distribution $b_t = [b_t(1), b_t(2), \ldots, b_t(|\mathcal{S}|)]$, where $b_t(i) = Pr[s_t = i|\bar{z}_t, \bar{a}_t]$. After observing an observation z, the agent updates its belief using the update function defined as $u_i(b, a, z, \Theta) = Pr[s_{t+1} = i \mid a_t = a, b_t = b, z_{t+1} = z, \Theta]$. Θ is the parameter for the underlying distributions for the POMDP, $\Theta = \{T, O, R\}$. The agent's policy π is a mapping from its belief to an action, and at each time step, the agent selects its action based on $a_t = \pi(b_t)$. The value function V depends on the policy π through the recursive equation known as the Bellman equation:

$$V^{\pi}(b, \Theta) = r(b, \pi(b), \Theta) + \gamma \sum_{z=1}^{|Z|} e_z(b, \pi(b), \Theta) V^{\pi}(u(b, \pi(b), z, \Theta), \Theta) \quad (2.7)$$

In Equation 2.7, $r(b, a, \Theta) = \sum_{s=1}^{|\mathcal{S}|} b(s)R(s, a)$ and $e_z(b, a, \Theta) = Pr(z_{t+1} = z \mid a_t = a.b_t = b, \Theta)$. The update operator $u(b, a, z, \Theta)$ represents the belief update after observing z, and it can be defined as:

$$u(b, a, z, \Theta) = [u_0(b, a, z, \Theta), u_1(b, a, z, \Theta,) \ldots, u_{|\mathcal{S}|}(b, a, z, \Theta)]$$

Each $u_i(b, a, z, \Theta)$ corresponds to $Pr(s_{t+1} = i \mid a_t = a, b_t = b, z_{t+1} = z)$.

The optimal value function is defined by the optimal Bellman equation:

$$V^{*}(b, \Theta) = \max_{a \in \mathcal{A}} \{r(b, a, \Theta) + \sum_{z=1}^{|Z|} e_z(b, \pi(b), \Theta) V^{\pi}(u(b, a, z, \Theta), \Theta)\} \quad (2.8)$$

The policy that acts greedily with respect to the optimal value function is the optimal policy:

$$\pi^*(b,\Theta) = \arg\max_{a\in\mathcal{A}} \left\{ r(b,\pi(b),\Theta) + \sum_{z=1}^{|Z|} e_z(b,a,\Theta) V^{\pi}(u(b,a,z,\Theta),\Theta) \right\}$$

This policy is known as the Bayes-optimal policy, as it strikes the optimal balance between exploration and exploitation. Exploration involves gathering more information to improve the policy, while exploitation involves maximizing rewards based on the current knowledge. By following the Bayes-optimal policy, the agent minimizes Bayesian regret, which measures the difference between the cumulative rewards obtained and the rewards that would have been obtained by always choosing the optimal action.

For the case of a finite-horizon bandit, the Bayesian cumulative regret is defined as follows:

Definition 1. *Let π be the agent's policy, b the belief about the environment, θ the true parameters of the environment, and T the horizon of the problem. In this context, $x^{\theta}_{\pi^*}$ denotes the reward for the optimal action in a bandit environment characterized by parameter θ, and $x^{\theta}_{\pi_t}$ represents the reward obtained by following policy π at time step t under the same environmental parameters. The Bayesian regret, which quantifies the performance of the policy π over the horizon T, is defined as:*

$$\textit{Bayesian regret} = \mathop{\mathbb{E}}_{\theta\sim b}[x^{\theta}_{\pi*} \times T - \sum_{t=1}^{T} x^{\theta}_{\pi_t}] \tag{2.9}$$

2.4.2 Bayes-Adaptive MDP in Multi-Armed Bandits

In the previous section, we introduced the Bayes-optimal policy in POMDPs. In this section, we explore the Bayes-optimal policy in Bayes-adaptive MDP (BAMDP) since it can be used in an MAB problem. BAMDP is a special case of the POMDP framework that incorporates the exploration-exploitation trade-off, in which the observation variable is the reward $z = x$ [7]. Solving the BAMDP provides an optimal solution to balancing exploration and exploitation.

Let $\mathcal{M}$ be the set of all MDPs. When the parameters of the underlying environment (MDP) are unknown, the Bayesian framework can represent the uncertainty about the MDP. This uncertainty can be incorporated into a probability distribution called the belief $b_t \in \mathcal{B}$, which captures the likelihood of each MDP $m \in \mathcal{M}$ corresponding to the environment at time step t.

The BAMDP is defined by the tuple $\langle \Omega, \mathcal{A}, T', R' \rangle$, where:

- $\Omega = \mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ is the set of all possible beliefs over $\mathcal{M}$.
- $\mathcal{A}$ is the set of actions (action space), and $\mathcal{A}_s$ is the set of actions available at state s.
- T' is the probability density function of the BAMDP's transition, which conditions the next state and belief on the current state, belief, and action. It can be defined as

  $$T'(s', b'|s, b, a) = \Pr(s_{t+1} = s', b_{t+1} = b'|s_t = s, b_t = b, a_t = a)$$

  Since the belief update function is deterministic, we can define the transition as

  $$\Pr(w_{t+1}|w_t, a) = \Pr(s_{t+1}, b_{t+1}|s_t, b_t, a_t)$$

  Note: s_t, b_t together form a Markov state, called the hyper-state w.
- R' is the probability density function of the reward for the BAMDP, which conditions the immediate reward on the current state, belief, and action. It can be defined as

  $$R'(r|s, a) = \Pr(r_{t+1} = r|s_t = s, b_t = b, a_t = a)$$

In this thesis, we assume that both the state space and action space are finite. The belief b can be any arbitrary distribution, and the rewards are real numbers, and its domain is known to the agent. In the case of a k-armed bandit setting, the state space $\mathcal{S}$ corresponds to the number of pulls left in the finite horizon or the number of pulls so far in the discounted infinite horizon. Figure 2.2 illustrates an example BAMDP, where the rectangles represent

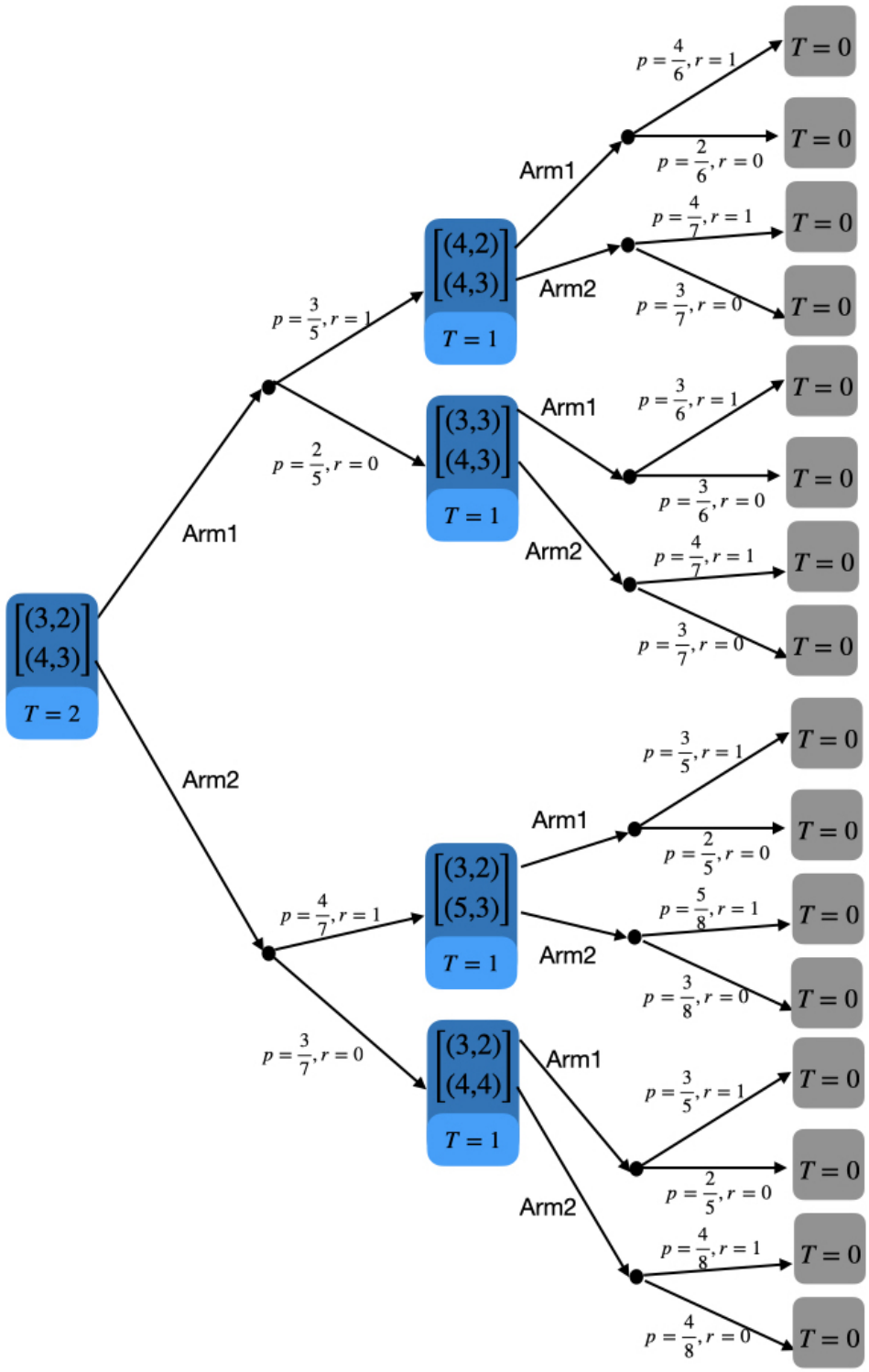

Figure 2.2: Example BAMDP: The rectangles represent belief-augmented states, with blue color indicating intermediate states and gray color indicating terminal states.

hyper-states, with blue color indicating intermediate states and gray color indicating terminal states.

Example 2.4.1. *Here is an example of a BAMDP (Figure 2.2) for a Beta-Bernoulli Multi-armed bandit with two arms. The prior belief b_0 at time step 0 is $\begin{bmatrix}(3,2)\\(4,3)\end{bmatrix}$ for each arm, and the horizon of the problem is $T = 2$.*

BAMDP provides a framework for finding the optimal exploration-exploitation policy. However, computing such a policy becomes computationally expensive as the environment grows larger.

2.5 Value of Information

This section delves into the concept of the value of information (VOI) and its various definitions. We begin by discussing the utility-based definition and then proceed to explore VOI in the context of POMDP and BAMDP. Additionally, we examine a specific definition in BAMDP called the Value of Current Sample Information (VOCI), which is utilized in Chapters 3 and 4. Finally, we investigate a myopic definition of VOI known as myopic-VOI.

2.5.1 Utility-Based Value of Information

The utility-based value of information was initially introduced as the "Value of Clairvoyance" or expected value of perfect information (EVPI) [11] and has since been further investigated by researchers such as Lawrence et al. [16] and Raiffa et al. [18].

Let us now define the utility-based value of information. Suppose $u_t(a, \theta)$ represents the measurable utility function of taking action a in experiment e with parameter θ at time t, where the action space is denoted as $A = \{1, \ldots, k\}$. Additionally, let $u_t(a_\theta, \theta)$ denote the maximum utility achievable from experiment e at time t, where $a_\theta = \arg\max_a(u(a, \theta))$. Furthermore, let b_t represent the belief (probability distribution) over the parameter θ of experiment e at time t. We can then define the conditional value of perfect information as:

$$VPI_\theta = u_t(a_\theta, \theta) - u_t(a', \theta)$$

In the above equation, θ denotes the perfect information (true parameter of experiment e), and a' represents the action chosen without the perfect information ($a' = \arg\max_a(\mathop{\mathbb{E}}_{\theta \sim b_t}[u(a, \theta)])$). Since the true parameter θ is unknown, we need to take the expectation of VPI over the belief b_t. In other words, EVPI is the agent's perception of the average VPI value. Therefore, the expected value of perfect information or $EVPI$ is given by:

$$EVPI = \mathop{\mathbb{E}}_{\theta \sim b_t}[VPI_\theta]$$

As evident, the Expected Value of Perfect Information ($EVPI$) quantifies the expected increase in utility resulting from knowledge of the perfect information regarding the experiment's true parameter.

2.5.2 Value Of Information in POMDPs

The concept of the value of information (VOI) focuses on the additional value gained in terms of long-term reward by permanently integrating a new information source into the problem setting. This idea has been explored in the field of infrastructure management engineering, specifically within the POMDP framework, by Srinivasan et al. [19] and Memarzadeh et al. [17]. The information considered in these studies refers to an added information source, typically a monitoring system, within the system.

Multiple definitions of VOI have been proposed in these works, and we will discuss two: (1) the value of the flow of information, which examines different methods of information availability for the agent, assuming the information source is accessible throughout all time steps until the terminal step; and (2) the value of the current information, which focuses solely on the value provided by the information at the present time step. For the subsequent steps, a policy for information availability can be assumed, such as assuming no information is available.

Here is a formal definition of the value of information. When referring to a POMDP with parameter θ, we consider a POMDP defined by the following tuple: $\langle \mathcal{S}, \mathcal{Z}, \mathcal{A}, \Theta = \{T, O, R\}, \mathbf{b}_0 \rangle$. Additionally, let $V^*(b, \Theta)$ denote the value function for the optimal exploration-exploitation strategy, which is equivalent to the Bayes-optimal policy discussed in Section 2.4. Now, we introduce two distinct POMDPs as follows:

- Θ: The parameter for the POMDP without the source of information.
- Θ': The parameter for the POMDP that incorporates an information source with a fixed availability method throughout all time steps (same availability method through all steps).

The value of the flow of information for a given belief state b, $VOI^f(b)$, is defined as:

$$VOI^f(b) = V^*(b, \Theta') - V^*(b, \Theta)$$

Depending on the chosen information availability method, different POMDPs (Θ') can be considered, resulting in various variants of the VOI^f definition. One example of an availability method is that the information source is available at each step with probability p.

2.5.3 Myopic Value Of Information in BAMDP

One variation of the value of information (VOI) concept is the myopic value of information. This notion focuses on the impact of information only for the immediate next step and is referred to as the Value of Perfect Information (VPI) [6]. To define VPI, we first introduce the concept of information gain, which quantifies the benefit the agent would obtain if it knew the reward for the next action a:

$$gain(x = x^a) = \begin{cases} [\mathbb{E}[X^{\alpha_2}] - x]^+ & \text{if } a = \alpha_1 \\ [x - \mathbb{E}[X^{\alpha_1}]]^+ & \text{if } a \neq \alpha_1 \end{cases} \tag{2.10}$$

In the above equation, α_1 represents the action suggested by the greedy policy, and α_2 denotes the second choice of action based on the greedy policy after α_1. The x^a is the observed reward after choosing action a.

The "gain" is a concept used to quantify the additional benefit an agent could accrue from knowing the reward for a forthcoming action, denoted as a. Gain is computed based on whether the action a is the one suggested by a greedy policy (α_1) or not. Let us assume that the agent chose to ask advice for arm $a = \alpha_1$. If the agent learns that the reward for the action $a = \alpha_1$ suggested by the greedy policy is not as beneficial as initially thought, it would then switch to the next best action (α_2). In this scenario, the gain is calculated as the difference between the expected reward of action α_2 and the actual reward for action $a = \alpha_1$, provided that this difference is positive. This is represented by the formula: $gain(x = x^a) = [\mathbb{E}[X^{\alpha_2}] - x]^+ \; a = \alpha_1$. On the other hand, let us assume that the agent chose to ask advice for arm $a \neq \alpha_1$. If the information about the reward for any other action suggests that this action is more rewarding than the action a suggested by the greedy policy (α_1), the agent would switch to that action. In this case, the gain is calculated as the difference between the reward for action a and the expected reward of action α_1, again only if this difference is positive. This is represented by the formula: $gain(x = x^a) = [x - \mathbb{E}[X^{\alpha_1}]]^+ \; a \neq \alpha_1$.

Now, let us define the VPI, which involves calculating the weighted average across all possible observable values for action a based on the agent's belief b:

$$VPI(b, a) = \mathop{\mathbb{E}}_{\substack{x \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [gain(x = x^a)] \tag{2.11}$$

2.6 Exploration Baseline policies

In the subsequent chapters, our primary focus is on Bayesian learning algorithms as we aim to approximate the Bayes-optimal policy discussed in Chapter 5. These algorithms are rooted in Bayesian learning and serve as baselines throughout this thesis. The following algorithms are used as baselines in our study: UCB, Thompson sampling (TS), Bayes-optimal policy, FH-Gittin, and

Bayesian Q-Learning.

2.6.1 Upper Confidence Bound

The Upper Confidence Bound (UCB) algorithm is rooted in the principle of optimism in the face of uncertainty. It suggests that the agent should act under the assumption that the environment provides the best possible reward. This algorithm is widely used in the domain of multi-armed bandit problems. UCB employs a specific action selection policy defined as:

$$A_t = \arg\max_a \left[Q_a + \underbrace{\sqrt{\frac{2 \log\, t}{N_a}}}_{\text{bonus}} \right] \tag{2.12}$$

Here, Q_a represents the unbiased estimate of the mean reward of arm a (μ_a), while N_a denotes the number of times arm a has been pulled until time-step t. The UCB algorithm uses a bonus-based approach to strike a balance between exploration and exploitation. This balance is achieved by assigning a "bonus" to each arm based on two factors: (1) the expected reward of the arm, as estimated from previous pulls, and (2) the uncertainty or variance in that estimate. The bonus term encourages the algorithm to explore arms with higher uncertainty, promoting exploration. At the same time, it favors arms with higher expected rewards, promoting exploitation. By adjusting the balance through the bonus term, the UCB algorithm aims to maximize the total reward accumulated over time. Notably, the UCB algorithm exhibits optimal asymptotic performance [15]. This implies that as the number of pulls (or rounds) increases indefinitely, the algorithm converges to the best possible expected reward.

2.6.2 Thompson Sampling

The Thompson sampling (TS) algorithm, a cornerstone in the realm of multi-armed bandit problems, embodies a distinctive approach to balance exploration and exploitation. Unlike the Upper Confidence Bound (UCB) algorithm, which leans on optimism in the face of uncertainty, Thompson Sampling

navigates the uncertainty landscape through a probability-matching mechanism, as it tries to pick an arm according to its optimal probability. This optimal probability is the likelihood, based on current belief, that a given arm will provide the best outcome among all available options [21]. Thompson sampling stands out for its elegance and simplicity in approximating the Bayes-optimal policy, making it a popular choice in various applications despite being only asymptotically optimal.

Thompson Sampling operates on a fundamentally different principle compared to UCB. Instead of adding a calculated bonus to encourage exploration, it leverages the underlying probability distributions to guide its choices, making it inherently probabilistic in nature. This probabilistic approach allows Thompson sampling to dynamically adjust its exploration-exploitation balance based on the evolving understanding of each arm's reward potential. While it may not match the exact performance of a Bayes-optimal policy, its ability to efficiently converge to optimal actions through sampling-based exploration makes it a robust and effective algorithm in practical scenarios.

The agent starts with a prior belief distribution over the parameters of the arms and the reward distribution according to that belief, called B_a and $Q(B_a)$, respectively. At each step, the agent will sample from its belief distribution and then pick the action greedily with respect to the sampled values.

Algorithm 1 Thompson Sampling Algorithm

Require: Prior distributions for the arms (b_0), problem setting horizon T

1: **for** $t = 1$ to T **do**
2: Sample arm parameters from the belief for all the arms: $\theta_t \sim b$ (with $\theta_{i,t}$ the parameter for arm i)
3: Select Arm $a_t = \arg\max_i \mathop{\mathbb{E}}_{\theta_{i,t}} [X^i]$
4: Observe reward $x_t^{a_t}$
5: Update the belief based on the observed reward $x_t^{a_t}$ (Section 2.5)
6: **end for**

This algorithm has been proven to have optimal asymptotic performance [15].

2.6.3 Bayes-Optimal Policy for Finite Horizon MAB

Let us assume that we are in the BAMDP (mentioned in Section 2.4.2) described as the tuple $\langle \Omega, \mathcal{A}, T', R' \rangle$. Our objective is to maximize the sum of rewards for a finite horizon T, and (X_t^π) is the value function for policy π defined by

$$V^\pi(w = (T, b)) = \mathop{\mathbb{E}}_{b} \left[\sum_{t=1}^{T} X_t^\pi \right]$$

The Bayes-optimal policy achieves the solution to Bayes' risk minimization [13]. Bayes' risk minimization is equivalent to solving the planning problem in the above BAMDP with a finite horizon criterion, that is, finding a policy that maximizes the $V^\pi(w = (T, b))$.

From learning theory [20], there exists a policy π^* that is optimal and its value function V^* satisfies the dynamic programming equation below:

$$\begin{aligned} V^*(w = (n, b)) &= \max_{a=1\ldots K} \Big(\mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [x^a] + \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [V^*(w = (n-1, b_{|x^a}))] \Big) \\ V^\pi(w = (0, b)) &= 0 \end{aligned} \tag{2.13}$$

The optimal value function can be calculated from the dynamic programming equation above (2.13) and the optimal policy π^* in state $w = (n, b)$, where n is the remaining time to play and b is the agent's belief, respectively, chooses an action $\pi(w = (n, b))$ that manage to reach the maximum in Equation 2.13. So, the optimal policy for the Bayesian MAB problem with a finite horizon is $a_t = \pi^*(w = (T - t + 1, b_{t-1}))$.

Unfortunately, due to the extremely large state space of Equation 2.13, the common methods to compute the optimal policy, such as dynamic programming, are often intractable. But, in some small scenarios, the optimal policy can be calculated and analyzed. The space of Equation 2.13 is extremely large because as we proceed into later steps, we build a large decision tree. The fan-out of this tree is equal to $|b_t| \times |T'|$, which is rather a large number; therefore, the decision tree is extremely large.

Example 2.6.1. *Here is an example of Bayes-optimal policy on top of problem setting in Example 2.4.1. Our problem-setting is Beta-Bernoulli multi-armed*

bandit with two arms, b_0 as prior belief ($b_0 = \begin{bmatrix}(3,2)\\(4,3)\end{bmatrix}$) for each arm, and the problem-setting horizon is $T = 2$. The dynamic programming equation (2.13) yields the following result for the optimal value function V^* (which depends on the b_0 and the horizon $T = 2$)

$$
\begin{aligned}
V^*(T=2, b_0) &= \max\{V_1, V_2\} \\
V_1 &= \mathop{\mathbb{E}}_{\substack{x^1 \sim Pr(.|\theta_1) \\ \theta_1 \sim b_1}} [x^1] \\
&\quad + P(x^1 = 1 \mid b_1) V^*(w = (T = 1, b_{|x^1=1})) \\
&\quad + P(x^1 = 0 \mid b_1) V^*(w = (T = 1, b_{|x^1=0})) \\
&= \frac{3}{5} + \frac{3}{5} \underbrace{\max\left(\frac{4}{6}, \frac{4}{7}\right)}_{\substack{\textit{Non-recursive} \\ \textit{since } T=1}} + \frac{2}{5} \underbrace{\max\left(\frac{3}{6}, \frac{4}{7}\right)}_{\substack{\textit{Non-recursive} \\ \textit{since } T=1}} \approx 1.23 \\
V_2 &= \mathop{\mathbb{E}}_{\substack{x^2 \sim Pr(.|\theta_2) \\ \theta_2 \sim b_2}} [x^2] \\
&\quad + P(x^2 = 1 \mid b_2) V^*(w = (T = 1, b_{|x^2=1})) \\
&\quad + P(x^2 = 0 \mid b_2) V^*(w = (T = 1, b_{|x^2=0})) \\
&= \frac{4}{7} + \frac{4}{7} \underbrace{\max\left(\frac{3}{5}, \frac{5}{8}\right)}_{\substack{\textit{Non-recursive} \\ \textit{since } T=1}} + \frac{3}{7} \underbrace{\max\left(\frac{3}{5}, \frac{4}{8}\right)}_{\substack{\textit{Non-recursive} \\ \textit{since } T=1}} \approx 1.19
\end{aligned}
$$

and as $V_1 > V_2$, the Bayes-optimal policy chooses arm 1."

2.6.4 Finite-Horizon Gittins' Index

An index-based policy in the MAB setting is a policy in which a numerical value can be assigned to each arm, given the state of the arms, independently of other arms, and then the arm with the greatest index is chosen. In Section 2.6.3 we talked about the Bayes-optimal solution. Gittin's theorem [10] says that the solution to Bayes-optimal policy for an infinite discounted setting with independent arms can be reduced to an index policy, later called Gittins indices. Moreover, the Gittins indices are only optimal in the discounted setting, but empirically, it is shown that they are near-optimal in a finite-horizon case [13]. In this thesis, we introduce Gittins' indices only in the context of (Bayesian) MAB models with independent arms for simplicity.

There are several interpretations of Gittins' indices, and one of the definitions relies on a calibration problem called the "one-armed bandit problem" for each arm [3] [13]. Let the "one-armed bandit problem" as $\mathcal{C}_\lambda$. Let $\theta \sim b$ and, conditionally on b, let (X_t) be an i.i.d. sequence with distribution v_θ, that is a one-armed bandit. For $\lambda \in \mathbb{R}$, we consider the following strategy, denoted by $\mathcal{C}_\lambda$. At each time-step t, the agent has to choose between receiving a (known) reward λ or drawing the unknown arm and receiving a random reward drawn from the distribution v_θ. The agent's goal is to maximize its rewards with respect to one of the criteria previously mentioned: either the sum of rewards for horizon T in a finite-horizon case or the sum of the discounted rewards in the infinite-horizon case.

Throughout the rest of this section, we focus on the finite-horizon case [14]. This strategy is a planning problem in an MDP, and by writing the dynamic programming equations for it, we can see that the optimal policy is a stopping policy. This stopping policy means that the unknown arm is played until some stopping time τ, which is a random variable, and after that, the known arm with reward λ is chosen until the end. One observation from $\mathcal{C}_\lambda$ is that as we increase the λ, the agent has less incentive to pull the unknown arm. There exists a critical value λ^* such that for a larger value of λ, the optimal policy for $\mathcal{C}_\lambda$ never draws the unknown arm. This critical value represents the price worth paying for playing the unknown arm, which is the Gittins' index.

In the finite-horizon one-armed bandit, the near-optimal policy plays the unknown arm as long as the current belief b on the parameter θ and the remaining time to pull an arm, n, are such that $G(n, b) > \lambda$, with $G(n, b)$ is the finite-horizon Gittins' index, defined as the following:

$$G(n, b) = \inf\{\lambda \in \mathbb{R} : \sup_{0 \leq \tau \leq n} \mathbb{E}_b[\sum_{t=1}^{\tau} X_t + \lambda(n - \tau)] = n\lambda\} \tag{2.14}$$

The definition of the Gittins' index from the original paper (in the discounted case only) is different than the definition above (2.14). It is not difficult to show that the following two definitions, 2.14 and 2.15, are the same

[9]. Here is the original definition of the FH-Gittins' index:

$$G(n,b) = \sup_{0<\tau\leq n} \frac{\mathop{\mathbb{E}}_{b}[\sum_{t=1}^{\tau} X_t]}{\mathop{\mathbb{E}}_{b}[\tau]} \tag{2.15}$$

The procedure for the computation of the FH-Gittins' indices is provided by E. Kaufmann [13]. In Beta-Bernoulli bandit, and for $n = 1, 2$, we can have an explicit expression of the Gittins' indices [4]. Here are these expressions:

$$G((\alpha,\beta),1) = \frac{\alpha}{\alpha+\beta} \text{ and } G((\alpha,\beta),2) = \frac{\alpha}{\alpha+\beta} \times \frac{1+\frac{\alpha+1}{\alpha+\beta+1}}{1+\frac{\alpha}{\alpha+\beta}} \tag{2.16}$$

In Equation 2.16, α and β are the Beta parameters of the arm that we are trying to calculate the FH-Gittins' index for.

Example 2.6.2. *Here is an example of the FH-Gittins policy on top of problem setting in Example 2.4.1. Our problem-setting will be Beta-Bernoulli multi-armed bandit with two arms, b_0 as prior belief ($b_0 = \begin{bmatrix}(3,2)\\(4,3)\end{bmatrix}$) for both arms, and the problem-setting horizon will be $T = 2$. From Equation 2.16, we have:*

$$G_1 = G((3,2),2) = \frac{3}{5} \times \frac{1+\frac{4}{6}}{1+\frac{3}{5}} = 1.6$$
$$G_2 = G((4,3),2) = \frac{4}{7} \times \frac{1+\frac{5}{8}}{1+\frac{4}{7}} \approx 1.46$$

and as $G_1 > G_2$, the FH-Gittin policy chooses arm 1. In this particular example, the Gittins policy is optimal, but generally, it is not.

2.6.5 Bayesian Q-learning

The Bayesian Q-learning algorithm is based on the idea of myopic VOI [6]. This algorithm uses myopic VOI as an incentive in a bonus-based policy, which essentially guides the selection of actions. This mechanism is somewhat reminiscent of the UCB policy; however, it employs myopic VOI as a bonus term added to the mean estimate of an arm's reward. The intuition behind this is to reflect not only the expected reward of an action but also the potential informational benefit associated with it. In particular, the myopic VOI provides

an estimate of the average gain in value that could be obtained if the agent had knowledge of the next step reward for a given arm. This added insight, encapsulated in the bonus term, steers the agent towards actions that are not only promising in terms of expected reward but also in terms of the informational value they carry. Here is the pseudo-code for the Bayesian q-learning algorithm:

Algorithm 2 Bayesian q-learning

Require: Prior distributions for the arms (b_0), problem setting horizon T

1: **for** $t = 1$ to T **do**
2: n = T-(t-1)
3: bonus^a =VPI(b$_{t-1}, a$)
4: Select Arm $a_t = \arg\max_a \left(\mathop{\mathbb{E}}_{b_{t-1}} [X^a] + \text{bonus} \right)$
5: Observe reward $x_t^{a_t}$
6: Update the belief based on the observed reward $x_t^{a_t}$ (Section 2.5)
7: **end for**

2.7 Conclusion

In this chapter, we explored the essential knowledge required to grasp the concepts explored in this thesis. This includes a thorough understanding of the Bayes-optimal policy in BAMDP as well as the notion of value of information (VOI). These foundational concepts and definitions are crucial for comprehending the advice problem-setting. Also, we showed that following the Bayes-optimal policy is intractable even in small problem settings, which yields the need for approximation. In Chapters 3 and 4, we develop a framework for formalizing the advice problem. Additionally, in Chapter 5, we explore various approximations and conduct analyses of those approximations for the Bayes-optimal policy.

Chapter 3

What Arm to Ask About

In this chapter, we investigate the problem of how to determine which arm an agent should seek advice about ("What arm to ask about problem"). First, we provide the optimal solution for this problem. Then, we further explore this problem within the context of an online finite-horizon Beta-Bernoulli multi-armed bandit with a specific type of advice: a single free sample from the arm selected by the agent as advice. In general, there are sometimes constraints on advice. For example, there may be a cost associated with each piece of advice or a budget for the number of times advice can be asked. In this work, we focus on advice with a budget. For the sake of simplicity and ease of analysis, throughout this chapter, we assume that the agent can only ask for advice once, and for the remaining steps, the agent follows the Bayes-optimal policy while interacting with the MAB in an online learning manner. Thus, the advice budget (Φ) is set to $\Phi = 1$. Additionally, we represent the time step horizon of the problem-setting as T. The type of advice mentioned above, which involves obtaining a sample from an arm, is related to the Bayes-optimal policy discussed in Chapter 5. This connection is one of the reasons we chose this type of advice and will be further explained in Section 3.3.

We need to mention that throughout the rest of this thesis, the problem setting will remain the same, online finite-horizon Beta-Bernoulli multi-armed bandit, with the Bernoulli parameters of arms' rewards being unknown to the agent.

3.1 Optimal Solution to the Problem of What Arm to Ask About

When the agent is seeking advice by observing a sample from a particular arm, one of the fundamental questions we encounter is, "What arm to ask about?" To answer this question, we need to assume a model for the advice. Given that we employ Bayesian learning, we maintain a belief distribution, represented by a Beta distribution in our case. This belief distribution incorporates the information acquired through the agent's interactions with the environment (multi-armed bandit). Since the true parameter of the environment is unknown, we assume that the advice model is based on the agent's belief. Let us assume that the agent has decided to seek advice at the current time step. The optimal arm for the agent to seek advice from is the arm that, on average, yields the highest return after seeking a piece of advice for that arm. This value is captured by the value of current sample information (VOCSI), which will be explained in detail in the next Section, 3.2.

3.2 Value of Current Sample Information

In Section 2.5 of the background chapter, we discussed the fundamentals of the value of information and its definition within a partially observable Markov decision process (POMDP) setting (Section 2.5.2). In this section, we present a derivation of the value function for a Bayesian adaptive Markov decision process (BAMDP) that we have developed. This value function is essential for defining the value of current information (VOCI) in BAMDPs. Subsequently, we investigate a derivation of the value of information in our specific setting (BAMDP), referred to as the value of current information (VOCI) in BAMDP. The VOCI will later be employed to define the value of current sample information (VOCSI), which incorporates a single sample from an arm as used in this thesis for advice.

3.2.1 Cross-action-Value Function vs. Action-Value Function in BAMDP

The cross-action-value function for BAMDP is useful when we want our policy to use a belief b' for decision-making that differs from the belief b used for the reward distribution. More specifically, at each step, the possible rewards are drawn from belief b, while the belief used in the policy to choose an action is b'. The equation for the action-value function, where the belief used for reward distribution is the same as the one used in the policy, is as follows:

$$\begin{aligned} q^*(w=(n,b),a) &= \mathop{\mathbb{E}}_{\substack{x^a\sim Pr(.|\theta_a)\\ \theta_a\sim b_a}}[x^a] + \mathop{\mathbb{E}}_{\substack{x^a\sim Pr(.|\theta_a)\\ \theta_a\sim b_a}}[V^*(w=(n-1,b_{|x^a}))] \\ q^*(w=(0,b),a) &= 0 \end{aligned} \tag{3.1}$$

The cross-action-value function, on the other hand, is given by the following:

$$\begin{aligned} q^{b'}(w=(n,b),a) &= \mathop{\mathbb{E}}_{\substack{x^a\sim Pr(.|\theta_a)\\ \theta_a\sim b_a}}[x^a] + \mathop{\mathbb{E}}_{\substack{x^a\sim Pr(.|\theta_a)\\ \theta_a\sim b_a}}[V^{b'_{|x^a}}(w=(k-1,b_{|x^a}))] \\ q^{b'}(w=(0,b),a) &= 0 \end{aligned} \tag{3.2}$$

In the above equations (3.1 and 3.2), "a" represents the arm chosen by the agent, and $V^{b'}(w=(n,b)) = q^{b'}(w=(n,b), \arg\max_a q^*(n,b',a))$ with the constraint $V^{b'}(w=(0,b))=0$. As you can see, there is a relationship between the optimal value function and the cross values function, as well as between the optimal action-value function and cross action-value function, which is depicted in the following:

$$\begin{aligned} V^b(w=(n,b)) &= V^*(w=(n,b)) \\ q^b(w=(n,b),a) &= q^*(w=(n,b),a) \end{aligned} \tag{3.3}$$

The two key equations in the BAMDP framework, Equations 3.1 and 3.2, serve different purposes:

- **Standard Action-Value Function** (Equation 3.1): Defines q^* for scenarios where the policy and reward distribution are based on the same belief b. It is essential for traditional decision processes in BAMDPs.

- **Cross-Action-Value Function** (Equation 3.2): Introduces $q^{b'}$ for cases where the decision-making belief b' differs from the reward distribution belief b.

Having both functions allows us to formulate the value of current information in a BAMDP in the upcoming Section 3.2.2.

3.2.2 Value of Current Information in BAMDP

The concept of the value of information, as discussed in Chapter 2.5.2, has been previously explored in the POMDP setting with predefined information structures. In this section, we extend this concept to the finite-horizon BAMDP setting, considering a general structure for information in the MAB problem-setting.

Let us consider a finite-horizon BAMDP with a horizon of n, described by the tuple $(\Omega, \mathcal{A}, T^{'}, R^{'})$. Let $V^*(w = (n, b))$ denote the value function for Bayes-optimal policy, where w is the hyper-state that represents the number of pulls left n and a belief b. The source of information is modeled by a random variable Ψ. When we observe a sample from this variable, denoted as $\psi \sim \Psi$ (receiving a piece of advice), the current belief state b is updated to b' using the update function $b' = b_{|\psi} = Pr(\cdot \mid \Psi = \psi)$. Moreover, the perfect information about an arm in an MAB setting is the value for the true underlying parameter of that arm.

Now, in order to define the value of current information (VOCI), first, let us define the gain as the increase in the value function by observing information for the current state, resulting in the current belief state b being updated to b':

$$\begin{aligned} gain(n, b, b') &= v^{b'}(w = (n, b')) - v^{b}(w = (n, b')) \\ &= v^{*}(w = (n, b')) - v^{b}(w = (n, b')) \end{aligned} \tag{3.4}$$

In the above Equation 3.4, we have used Equation 3.3 to transition from the first line to the second line.

Next, let us define the value of current information (VOCI) with information modeled by the random variable Ψ:

$$VOCI(n, b, \Psi) = \mathop{\mathbb{E}}_{\psi \sim \Psi}[gain(n, b, b_{|\psi})] \tag{3.5}$$

As you can observe, there is an analogy between the gain and the VOCI definition in this section and the utility-based VOI definition from Section 2.5.1. Gain is similar to the value of perfect information (VPI), and VOCI is similar to the expected value of perfect information (EVPI). Moreover, the notion of VOCI used here is analogous to VOI^f from Section 2.5.2. VOI^f defines the current value of information while assuming that no additional information is available for the remaining steps.

3.2.3 Value of Current Sample Information in BAMDP

Once the VOCI with the general model Ψ for information is defined, we can explore a specific information model that represents a free sample from an action a. The VOCI notion that uses a free sample from an action is the Value of Current Sample Information (VOCSI). More specifically, the information structure is defined as $\Psi_a = Pr(X^a|b_a)$, indicating that the agent's obtained information is equivalent to taking action a, observing the resulting reward, and subsequently updating its belief based on that acquired information. Let us define the VOCSI as follows:

$$\begin{aligned}
VOCSI(n, b, a) &= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[gain(n, b, b_{|x^a})] \\
&= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[v^*(w = (n, b_{|x^a})) - v^b(w = (n, b_{|x^a}))] \\
&= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[v^*(w = (n, b_{|x^a}))] - \underbrace{\mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[v^b(w = (n, b_{|x^a}))]}_{claim1 := V^*(w=(n,b)))} \\
&= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[v^*(w = (n, b_{|x^a}))] - v^*(w = (n, b))
\end{aligned} \tag{3.6}$$

Claim 1.

$$\mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[q^b(n, b_{|x^a}, a^*(n, b))] = q^*(n, b, a^*(n, b)) = v^*(n, b)$$

In the above equation, $a^*(n, b) = \arg\max_a q^*(n, b, a)$.

Proof. Proof of the above Claim 1 is provided in the appendix (Section A.1). □

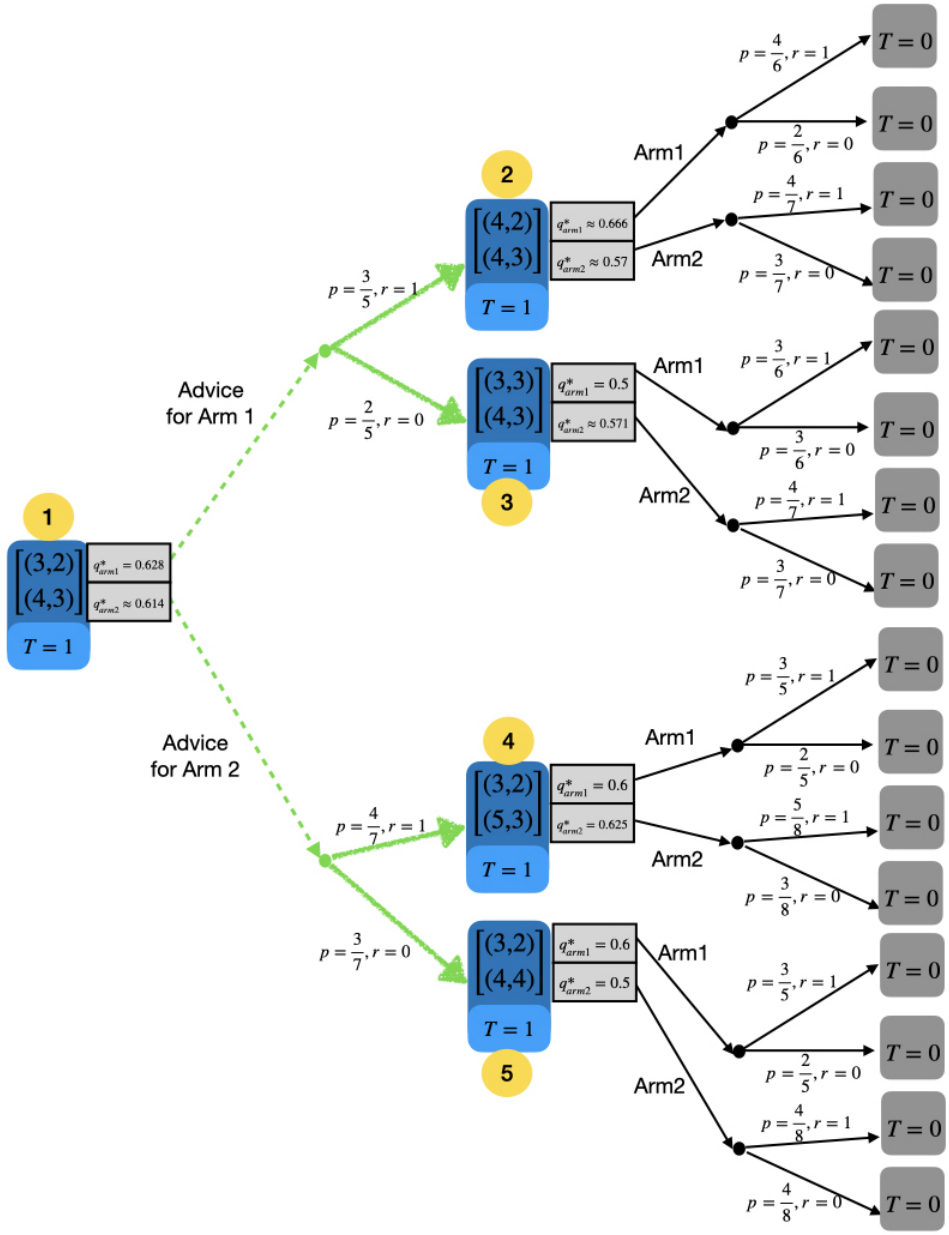

Figure 3.1: Example of a BAMDP for the "What arm to ask about problem." The rectangles represent belief-augmented states, with the blue color indicating intermediate states and the gray color representing terminal states. The yellow circle numbers the states, and the gray rectangle provides the optimal action value for their adjacent state and a specified arm.

Example 3.2.1. *Here is an example of a BAMDP for the optimal "What arm to ask about algorithm," illustrated in Figure 3.1. The example considers a Beta-Bernoulli multi-armed bandit with two arms, where b_0 represents the prior belief for each arm.*

$$b_0 = \begin{bmatrix}(3,2)\\(4,3)\end{bmatrix}$$

The horizon of the problem-setting is $T = 1$. From the optimal solution to the "What arm to ask about problem" (Section 3.1), we know that the optimal solution to the "What arm to ask about problem" is $\arg\max_a$ *VOCSI(n, b, a). Based on Equation 3.6, we can simplify the expression for VOCSI by ignoring the second term since it doesn't depend on action a. Thus,*

$$\arg\max_a \ VOCSI(n,b,a) = \arg\max_a \underset{\substack{x^a \sim Pr(.|\theta_a)\\ \theta_a \sim b_a}}{\mathbb{E}} [v^*(w = (n, b_{|x^a}))]$$

From Figure 3.1, we can observe the BAMDP representation of the below expression.

$$\underset{\substack{x^a \sim Pr(.|\theta_a)\\ \theta_a \sim b_a}}{\mathbb{E}} [v^*(w = (n, b_{|x^a}))]$$

The green lines in Figure 3.1 are associated with asking for advice for different arms in the current step.

The values of the expression $\underset{\substack{x^a \sim Pr(.|\theta_a)\\ \theta_a \sim b_a}}{\mathbb{E}} [v^*(w = (n = 1, b_{0|x^a}))]$ *for different arms are shown in the gray box adjacent to state number 1. As can be seen, the value for arm 1 is greater than that for arm 2, indicating that* $\arg\max_a$ *VOCSI(n, b, a) = arm 1.*

3.3 Bayes-Optimal Policy vs. VOCSI

In this section, we discuss the close relationship between the Bayes-optimal policy and VOCSI in a finite-horizon Bayesian setting. Let $q^*(n, b, a)$ be the action value function of action a for the Bayes-optimal policy and let *VOCSI*(n, b, a) be the value of current sample information for action a given belief b and horizon n. We claim that the relation between these two is as follows:

Claim 2.

$$q^*(n+1,b,a) = \mathop{\mathbb{E}}_{b}[X^a] + VOCSI(n,b,a) + v^*(n,b)$$

Proof. From Equation 2.13, we have:

$$\begin{aligned} q^*(n,b,a) &= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[x^a] + \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[\max_{a'} q^*(n-1, b_{|x^a}, a')] \\ &= \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[x^a] + \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}}[v^*(n-1, b_{|x^a})] \end{aligned} \tag{3.7}$$

$$\xrightarrow{\textit{Equation 3.7 \& 3.6}} \begin{aligned} q^*(n+1,b,a) =& \mathop{\mathbb{E}}_{b}[X^a] + VOCSI(n,b,a) + \max_{a'} q^*(n,b,a') \\ =& \mathop{\mathbb{E}}_{b}[X^a] + VOCSI(n,b,a) + v^*(n,b) \end{aligned}$$

□

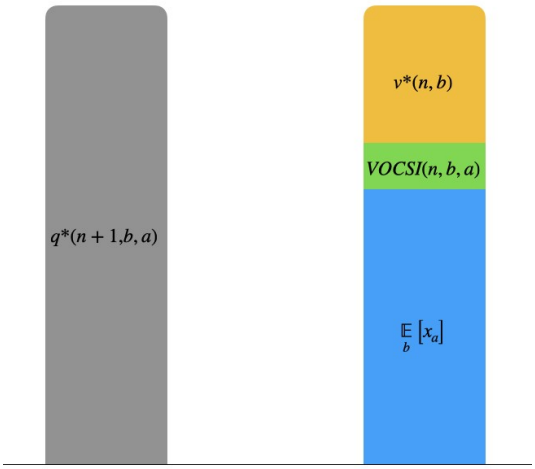

Figure 3.2: Bayes-optimal value function decomposition visualization. The gray bar represents the action value for arm a for the Bayes-optimal policy with horizon $n+1$ and belief b. The blue bar represents the expected value of arm a, the green bar represents the VOCSI value for arm a with horizon n and belief b, and the yellow bar represents the value function of the Bayes-optimal policy with horizon n and belief b.

As shown in Claim 2, the action value function for the Bayes-optimal policy with horizon $n+1$ can be decomposed into three terms. The first term represents the expected value of action a. The second term represents the value of current sample information (VOCSI) for action a with horizon n and

belief b. These first two terms are dependent on the chosen action a. Finally, the third term represents the value function for the Bayes-optimal policy with horizon n, which is independent of the action chosen. Figure 3.2 visualizes this decomposition.

Based on Claim 2, we can provide an equivalent action selection policy to the Bayes-optimal exploration strategy (Claim 3). In this equivalent policy, we consider only the first two terms on the right-hand side of Claim 2 since they are the only terms dependent on the action.

Claim 3. *If a^* is the action selected by the Bayes-optimal policy at state (n, b): $a^* = \arg\max_a q^*(n, b, a)$, we claim that the following policy provides the same best action selection policy as a^*:*

$$a^* = \arg\max_a \left(\underset{b}{\mathbb{E}}[X^a] + VOCSI(n-1, b, a)\right) \tag{3.8}$$

Proof.

$$\begin{aligned} a^* &= \arg\max_a q^*(n, b, a) \\ \xrightarrow{Claim\ 2} &= \arg\max_a \left(\underset{b}{\mathbb{E}}[X^a] + VOCSI(n-1, b, a) + \underbrace{\max_{a'} q^*(n, b, a')}_{\text{independent of a}}\right) \\ &= \arg\max_a \left(\underset{b}{\mathbb{E}}[X^a] + VOCSI(n-1, b, a)\right) \end{aligned}$$

□

3.4 Approximation of the Optimal Solution to What Arm to About

The optimal solution presented in Section 3.1 becomes intractable when problem-setting parameters are increased, such as the number of arms or the horizon. Therefore, it is necessary to propose approximations for this problem in order to reduce the computational cost of optimal methods. In the next two sections, we introduce two types of approximations, which are the h-myopic approximation and the Gittins' argmax approximation of VOCSI alongside

with their pros and cons. Additionally, in Section 3.4.1, we explore the myopic approximation used in the literature and its relationship to the h-myopic approximation.

H-myopic Approximation

One approach to approximate the optimal solution to the "What arm to ask about problem" is by approximating the problem-setting horizon. We refer to this approximation type as the "h-myopic" approximation, with h representing the approximation horizon. For example, when we mention "3-myopic VOCSI," we assume the algorithm's horizon is three. Specifically, we calculate the values based on the assumption that the problem-setting horizon is $h = 3$, even though the problem-setting's true horizon may differ. For the "What arm to ask about problem," when approximating the VOCSI, the myopic approximation is as follows:

- Constant horizon: In this horizon approximation, we assume that the horizon is a fixed number and does not decrease as we progress. For instance, if we use a 3-myopic approximation, $VOCSI(n, b, a)$ is approximated by $VOCSI(3, b, a)$ for all n.

This myopic approximation enables us to obtain an approximate value for VOCSI, which can then be used to determine the argmax approximation of VOCSI by taking argmax over the approximated values. Here are the pros and cons of h-myopic approximation of VOCSI:

Pros:

- **Value Estimation**: Offers an estimated value for VOCSI, useful in scenarios requiring a comprehensive understanding of VOCSI, such as when evaluating costs associated with asking for advice.

- **Customizable Horizon**: Allows for setting a specific horizon, providing flexibility in balancing foresight and computational resources.

Cons:

- **Less Efficient for Equivalent Horizon**: Slower compared to Gittins' argmax approximation when using the same horizon, leading to higher computational costs.

- **Accuracy Dependent on Horizon Length**: May not adequately capture long-term effects or the dynamics of the decision-making environment if the horizon is set too short.

However, there is another type of approximation that can only provide us with the argmax approximation of VOCSI, known as "Gittins' argmax approximation of VOCSI." The "Gittins' argmax approximation of VOCSI" is introduced in the following section.

Gittins' Argmax Approximation of VOCSI

In Section 2.6.4, we discussed the FH-Gittins' index, a technique that provides a policy close to optimal when compared to the Bayes-optimal policy. This closeness to optimality will be empirically demonstrated in the upcoming optimal exploration chapter (5.3.3). Consequently, it is reasonable to approximate $\textit{FH-Gittins}^{M}_{index} \approx \arg\max_a q^*(n, b, a)$, where M corresponds to the underlying MDP for the $q^*(n, b, a)$ optimal action-value function. Considering the nature of the advice in VOCSI and its relation to the Bayes-optimal policy (Section 3.3), we propose applying the FH-Gittins technique to an augmented MDP (M'), which will be described shortly, to approximate the argmax of VOCSI. Based on Claim 2, we have $VOCSI(n, b, a) = q^*(n+1, b, a) - \mathop{\mathbb{E}}_{b}[X^a] - v^*(n, b)$; therefore, $\arg\max_a \textit{VOCSI}(n, b, a) = \arg\max_a (q^*(n+1, b, a) - \mathop{\mathbb{E}}_{b}[X^a])$. Now, let us define an augmented MDP M' such that its optimal action-value function ($q^{'*}$) is given by $q^{'*}(n+1, b, a) = q^*(n+1, b, a) - \mathop{\mathbb{E}}_{b}[X^a]$. To achieve this, we modify the MDP M such that the starting state's action reward is zero. By applying the FH-Gittins' technique to this augmented MDP, we obtain an approximation denoted as $\textit{FH-Gittins}^{M'}_{index}$, which can be used as an approximation for $\arg\max_a \textit{VOCSI}(n, b, a)$. Here are the pros and cons of Gittins' argmax approximation of VOCSI:

Pros:

- **Optimal Action Focus**: Excellently suited for scenarios primarily concerned with identifying the best action, due to its focus on the argmax of VOCSI.

- **Greater Efficiency**: Significantly faster than the h-myopic approximation, especially valuable when dealing with the same horizon lengths.

Cons:

- **Limited to Action Selection**: Does not provide an actual value estimate for VOCSI, making it less suitable for scenarios where understanding the full value of VOCSI is necessary.

The choice between these two approximations hinges on the specific needs of the decision-making process. If the goal is to understand the full value of VOCSI, especially when additional factors like advice cost are in play, the h-myopic approximation is more appropriate despite its computational intensity. In contrast, for scenarios where quick and efficient identification of the best action is paramount, Gittins' argmax approximation is the superior choice, offering significant speed advantages, particularly at the same horizon lengths. The decision should be guided by the balance between the need for comprehensive value analysis and the efficiency of action selection.

3.4.1 Myopic Value Of Information

In the background (Section 2.5.3), we discussed myopic VOI (VPI), and now let us explore its relation to $VOCSI$. According to Claim 4 below, the VPI notion (Equation 2.11) is equivalent to 1-myopic VOCSI.

Claim 4. *If b represents the agent's current belief and a denotes an arm, then:*

$$\textit{1-myopic } VOCSI(b,a) = VOCSI(n=1,b,a) = VPI(b,a)$$

Proof. As we can observe, the expectation in Equation 2.11:

$$VPI(b,a) = \mathop{\mathbb{E}}_{\substack{x\sim Pr(.|\theta_a)\\ \theta_a\sim b_a}}[gain(x=x^a)]$$

is the same as the second line of Equation 3.6:

$$VOCSI(n, b, a) = \mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [v^*(w = (n, b_{|x^a})) - v^b(w = (n, b_{|x^a}))]$$

Therefore the proof is complete if the expressions inside these expectations are equal.

Now, let us compare Equation 2.10:

$$gain(x = x^a) = \begin{cases} [\mathbb{E}[X^{\alpha_2}] - x]^+ & \text{if } a = \alpha_1 \\ [x - \mathbb{E}[X^{\alpha_1}]]^+ & \text{if } a \neq \alpha_1 \end{cases}$$

to the expression inside the expectation in the second line of Equation 3.6:

$$v^*(w = (n, b_{|x^a})) - v^b(w = (n, b_{|x^a}))$$

As indicated in Equation 2.10, when we obtain perfect information from the action suggested by the greedy policy, the gain is the difference between the expected value of the second greedy action and the reward obtained from the information. Additionally, if the gain is negative, we set the gain to zero since we won't switch to the second greedy action. Therefore, if we determine that the true value of the second greedy action is higher than that of the highest greedy action, we choose the second greedy action instead. It can be observed that $[\mathbb{E}[X^{\alpha_2}] - r]^+$, where $a = \alpha_1$, is equal to $v^*(w = (n, b_{|x^a})) - v^b(w = (n, b_{|x^a}))$, where $\mathbb{E}[X^{\alpha_2}] = v^*(w = (n, b_{|x^a}))$ and $r = v^b(w = (n, b_{|x^a}))$. Similarly, suppose we obtain perfect information from the action not suggested by the greedy policy. In that case, the gain is the difference between the reward obtained from the information and the expected value of the greedy action. Therefore, if we receive information indicating that an action other than the greedy action has a higher value, we switch to that action. In doing so, it can be observed that $[r - \mathbb{E}[X^{\alpha_1}]]^+$, where $a \neq \alpha_1$, is also equal to $v^*(w = (n, b_{|x^a})) - v^b(w = (n, b_{|x^a}))$, where $r = v^*(w = (n, b_{|x^a}))$ and $\mathbb{E}[X^{\alpha_1}] = v^b(w = (n, b_{|x^a}))$.

□

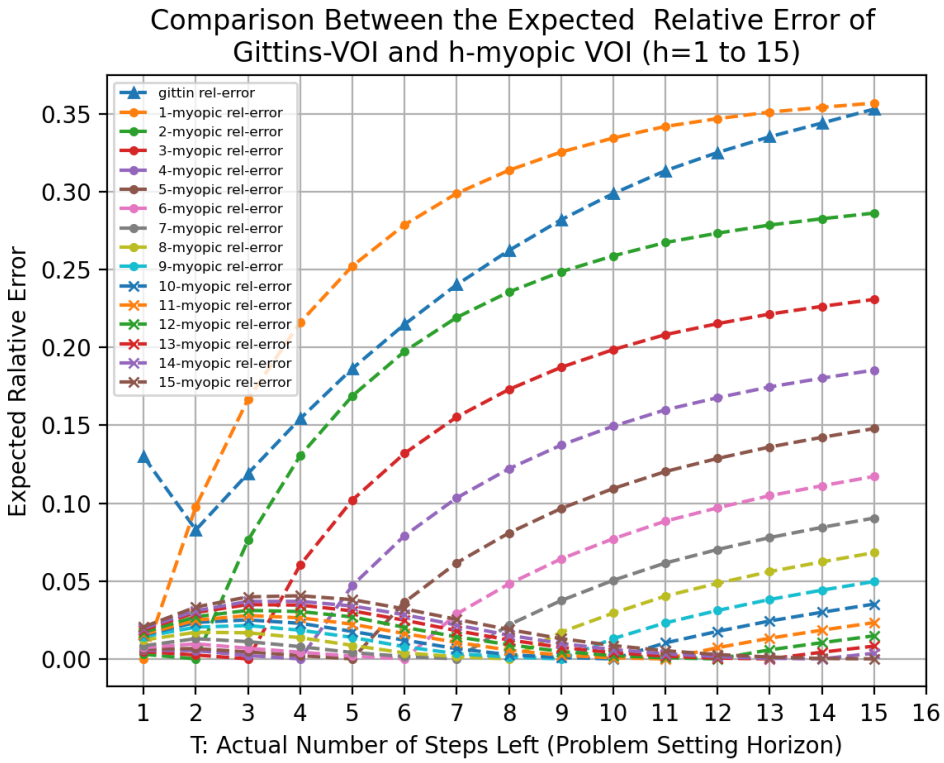

Figure 3.3: Relative error of Gittins' argmax approximation and h-myopic approximation for $i = \{1, \ldots, 15\}$. The relative error is computed as the difference between the optimal solution value and the approximation solution value, divided by the optimal solution value.

3.5 Experiments

In this section, we outline the experimental design (Section 3.5.1) and provide an analysis of the results (Section 3.5.2).

3.5.1 Design

The experimental evaluation compares the performance of the optimal approach and the approximation approach for the "What arm to ask about" algorithm, as described in Section 3.4. We conduct two sets of experiments:

- **Optimal solution:** We evaluate the performance of the optimal approach for the "What arm to ask about problem."

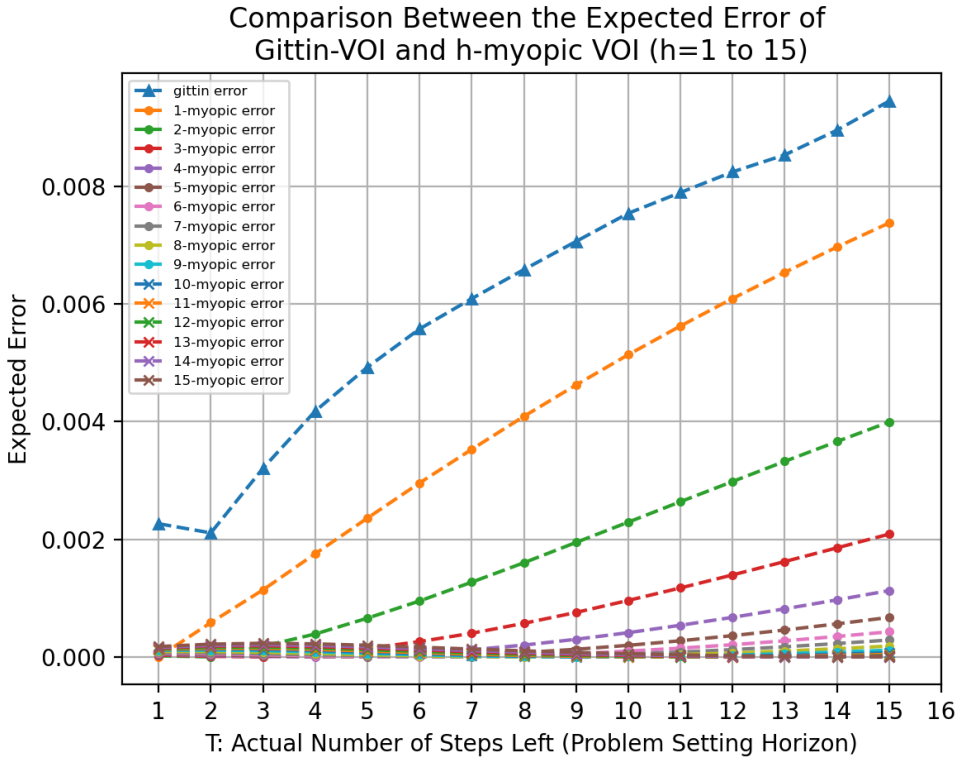

Figure 3.4: Absolute error of Gittins' argmax approximation and h-myopic approximation for $i = \{1, \ldots, 15\}$. The absolute error is computed as the difference between the optimal solution value and the approximation solution value.

- **Approximate solution:** We assess the effectiveness of the approximation approach for the "What arm to ask about problem."

For ease of analysis in the subsequent section (Section 3.5.2), we use a Beta-Bernoulli multi-armed bandit model with three arms, each having a uniform prior belief. Furthermore, we vary the initial beliefs over a range of values (10^6 different values) $B = \left\{ \begin{bmatrix} (\alpha_1, \beta_1) \\ (\alpha_2, \beta_2) \\ (\alpha_3, \beta_3) \end{bmatrix} \middle| \forall \alpha_i, \beta_i \in \{1, \ldots, 10\} \right\}$ to obtain more accurate and reliable results.

3.5.2 Analysis

In this section, we conduct an analysis of the "What to ask about problem" using different types of approximations as discussed in the previous section

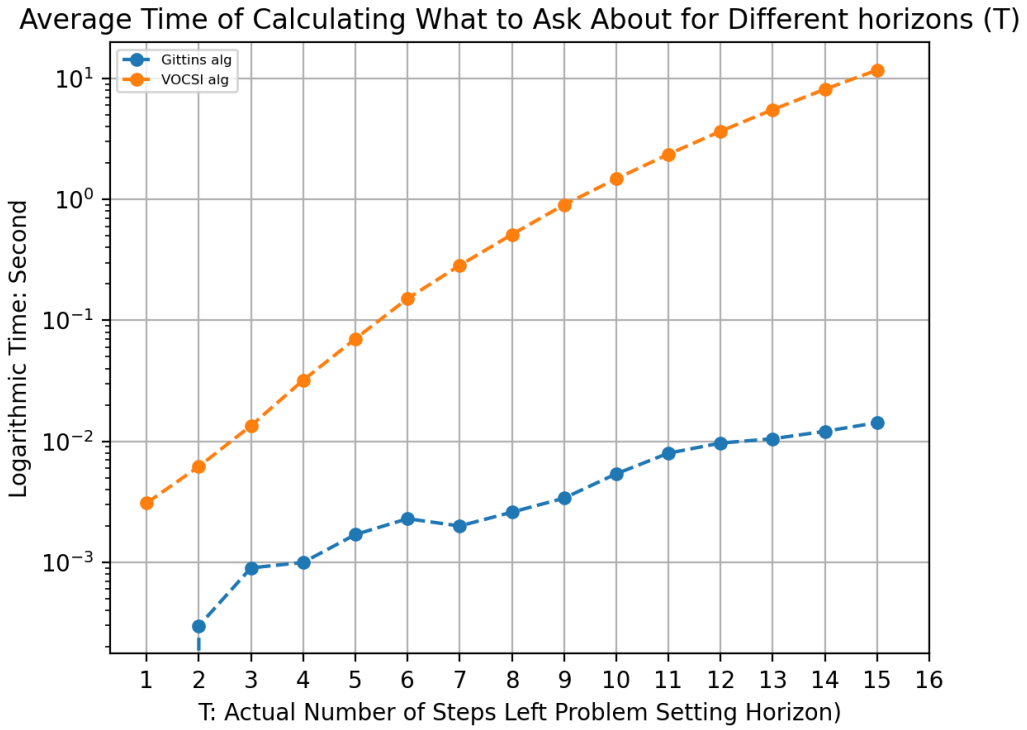

Figure 3.5: Average time taken by Gittins' argmax and VOCSI algorithm to determine which arm to ask for advice, based on the problem-setting horizon (T). These results were run on an "11th gen intel(R) core(TM) i7-11700F @ 2.5GHz" CPU.

(Section 3.4). All results discussed are averaged over the belief space, consisting of 10^6 different initial beliefs, as outlined in Section 3.5.1. It is important to note that due to the large number of initial beliefs ($N = 10^6$), error bars are omitted since the errors are negligible.

Figure 3.3 illustrates the relative error of the h-myopic approximation and Gittins' argmax approximation. Notably, the 1-myopic approximation serves as a baseline, and it is equivalent to the myopic approximation (Section 3.4.1). **This result shows that increasing the approximation horizon leads to a lower relative error. Importantly, if we overestimate the approximation horizon (selecting a horizon where $h > T$), the performance is significantly better compared to underestimating it.** Additionally, the performance of "Gittins' argmax approximation of VOCSI" initially out-

performs the 1-myopic algorithm. However, as the number of steps increases, the relative error of "Gittins' argmax approximation of VOCSI" approaches, and at times exceeds, that of the 1-myopic algorithms. This can be attributed to the sub-optimality of the Gittins' index policy when used in a finite-horizon setting, as it is optimally designed for the discounted setting. Consequently, this inherent limitation of the Gittins' algorithm leads to the sub-optimality of the "Gittins' argmax approximation of VOCSI."

Figure 3.4 presents the absolute error of the h-myopic approximation and "Gittins' argmax approximation of VOCSI." In terms of absolute error, the performance of the "Gittins' argmax approximation of VOCSI" is inferior to that of the h-myopic approximation. Therefore, **based on the results depicted in Figures 3.3 and 3.4, we conclude that the h-myopic approximation outperforms the Gittins' argmax approximation.**

Lastly, Figure 3.5 provides a temporal comparison between the Gittins' argmax algorithm and the VOCSI algorithm for different horizons. The y-axis of the figure uses a logarithmic scale in seconds, revealing that both algorithms exhibit exponential growth with respect to the horizon. However, the VOCSI algorithm demonstrates a significantly faster growth rate compared to the Gittins' argmax algorithm. Note that the Gittins' argmax algorithm lacks a data point at horizon $T = 1$. This is attributed to numerical errors, which led to a recorded time of zero. As the logarithm of zero is negative infinity, this point is absent.

3.6 Conclusion

In this chapter, we examined the optimal solution to the "What arm to ask about problem" and introduced two types of approximations for addressing this problem: the h-myopic algorithm and the Gittins' argmax algorithm. The h-myopic algorithm approximates the problem horizon, while the Gittins' VOCSI algorithm uses the finite-horizon Gittins' technique to estimate the maximum value of the VOCSI term across the action space. We then conducted an analysis of the optimal solution and its approximations, leading to

several key findings.

Firstly, the study reveals that increasing the approximation horizon leads to notable reductions in both relative and absolute errors. This implies that a more precise estimation of the problem horizon plays a vital role in enhancing decision-making accuracy and improving the quality of approximations.

Additionally, the research emphasizes the critical importance of appropriately selecting the approximation horizon. Surprisingly, overestimating the horizon, even beyond the true problem horizon, exhibits remarkably superior performance compared to underestimating it. This highlights the necessity of considering an extended horizon that accounts for future uncertainties and optimizes decision-making effectively.

The comparison between the Gittins' argmax algorithm and 1-myopic algorithms yields intriguing observations. Initially, Gittins' argmax algorithm outperforms the 1-myopic algorithm in terms of relative error. However, as the number of steps increases, the relative error of Gittins' argmax algorithm gradually converges to and even surpasses that of the 1-myopic algorithms. This phenomenon can be attributed to the finite-horizon setting's limitations in leveraging the full potential of the Gittins' index policy, which is primarily optimized for the discounted setting. Consequently, the sub-optimality of the Gittins' algorithm influences the overall performance of the Gittins' argmax approximation.

Furthermore, the evaluation of absolute error indicates that the Gittins' argmax algorithm performs less favorably than the h-myopic approximation. Based on the assessment of both relative and absolute errors depicted in Figures 3.3 and 3.4, a clear conclusion can be drawn: in expectation the h-myopic approximation consistently outperforms the Gittins' argmax approximation in our problem-setting.

Chapter 4

When To Ask For Advice

In this chapter, we examine the circumstances under which an agent should seek advice, focusing on the problem setting introduced in the previous chapter (Chapter 3). Firstly, we present the optimal solution to the "When to ask for advice problem"(Section 4.1). Next, we explore various approximations for the optimal solution (Section 4.2). Finally, we conduct experiments and analyze the results (Section 4.3).

4.1 Optimal Solution to the Problem of When to Ask for Advice

When considering the scenario of an agent seeking advice, a fundamental question arises: "When should the agent ask for advice?" More specifically, should I ask for advice right now or postpone? To the best of our knowledge, this question has not been thoroughly examined. Some studies have used approximations based on the agent's uncertainty to determine whether advice should be sought [5] or they decide to provide advice if there is a substantial difference between the action values of the best and worst actions [1], [22]. However, these approaches do not account for the optimal time step for seeking advice; they merely offer approximations. To address this question, we leverage the "What arm to ask about algorithm" introduced in Chapter 3.

Suppose the agent has $n = T \in \mathbb{N}$ remaining time steps (T is problem setting's horizon), and it must determine whether to ask for advice immediately or delay the request until a later step. To identify the optimal time

step for seeking advice, the agent calculates the expected maximum gain by postponing advice for $t < T$ steps into the future. To compute this gain, the agent evaluates the expected maximum of the current sample information's value (VOCSI) across all potential future belief states achievable after t steps. Then, the agent repeats this process for all possible time steps $t \in \{1, \ldots, T\}$ at which it can postpone the request, then selects the time step that yields the maximum expected gain. If this time step is not the current time step, the agent should defer asking for advice. Otherwise, it should seek advice immediately. The corresponding pseudo-code for the Optimal "When to Ask Algorithm" is provided below (see Algorithm 3). The optimal solution for the "When to ask for advice problem"is not limited to any specific type of advice. As long as the advice structure can be modeled using the format: $b' = b_{|\psi} = Pr(\cdot \mid \Psi = \psi)$ with ψ be the advice, we can use VOCI instead of VOCSI and generalize the advice type.

Algorithm 3 Optimal When to Ask Algorithm

Require: Prior distributions for the arms (b_0), problem setting horizon T, number of arms $|\mathcal{A}|$, Bayes-optimal policy (π^*), and $R(n, b, a)$ is the stochastic reward function

Initialize: $EVOI_{list} \leftarrow$ empty list, $B_{list} \leftarrow [b_0]$
for $t = 1$ to T **do**
    $n = T - (t - 1)$
    // Initialize VOI^n_{list} as an empty list for storing VOI values at time step n
    $VOI^n_{list} \leftarrow$ empty list
    for each belief state b in B_{list} **do**
    // Compute the maximum VOCSI for each action a and belief state b
        $VOI_{max} = \max_a VOCSI(n, b, a)$
        $VOI^n_{list}.append(VOI_{max})$
    end for
    // Calculate the expected VOI for time step n
    $evoi_n \leftarrow \mathbb{E}[VOI^n_{list}]$
    // Initialize B^{next}_{list} as an empty list for next belief states
    Initialize: $B^{next}_{list} \leftarrow$ empty list
    for each belief state b in B_{list} **do**
    // Choose the best action a using Bayes-optimal policy $\pi^*(n, b)$
        Select Arm $a = \pi^*(n, b)$
        for each possible reward x^a for action a **do**
    // Update the belief state $b' = b_{|x^a}$
            $b' = b_{|x^a}$
            $B^{next}_{list}.append(b')$
        end for
    end for
// Set B_{list} to B^{next}_{list} for the next iteration
    $B_{list} \leftarrow B^{next}_{list}$
end for
// Determine the time step to wait, $wait_{index}$, using the maximum value in $evoi_n$
$wait_{index} = \arg\max evoi_n$
// Return decision to Ask for advice if the $wait_{index}$ is more than one step
if $wait_{index} \neq 1$ **then**
    return Ask
else
    return $Wait$
end if

Claim 5. *In the context of a Bayesian decision-making process, where an agent seeks to maximize its expected cumulative reward over a decision horizon T,*

optimizing the Expected Value of Current Sample Information (E[VOCSI]) at each decision step maximizes the expected cumulative return. Specifically, if the agent, at each time step t, chooses to seek advice based on the maximization of E[VOCSI], then this strategy will lead to an optimal sequence of decisions that maximizes the overall expected return from the current time step to the end of the decision horizon.

Proof. Proof of the above Claim 5 is provided in the appendix (Section A.2). □

One intriguing aspect of our findings is the optimality of deferring advice-seeking in certain scenarios, even in scenarios where obtaining advice incurs no cost but there is only a budget. This observation, at first glance, appears counterintuitive, as one might naturally assume that free advice should be sought immediately to maximize information gain without any trade-offs.

In certain situations, the optimal strategy involves waiting for a more opportune moment to seek advice, despite its cost-free nature. This can be attributed to several factors. Firstly, the value of advice may not be uniformly distributed across all time steps; it might be more beneficial at specific junctures due to the evolving state of the system or the agent's knowledge. Secondly, seeking advice, even when free, can still alter the decision-making landscape. It might lead to different paths or choices that could be less advantageous than those taken without immediate advice. Finally, in a dynamic environment, the information's relevance and utility can change over time, making the timing of advice a crucial factor.

Thus, the decision to delay seeking free advice is not merely a function of its cost but a strategic consideration of its timing and potential impact on the overall decision-making process. In the following, you can find an example of such a phenomenon, in which it is beneficial to postpone the advice.

Example 4.1.1. *In Figure 4.1, we present an illustrative example of an optimal "When to ask for advice algorithm" applied to a Beta-Bernoulli Multi-armed bandit problem. The problem consists of three arms, with the prior*

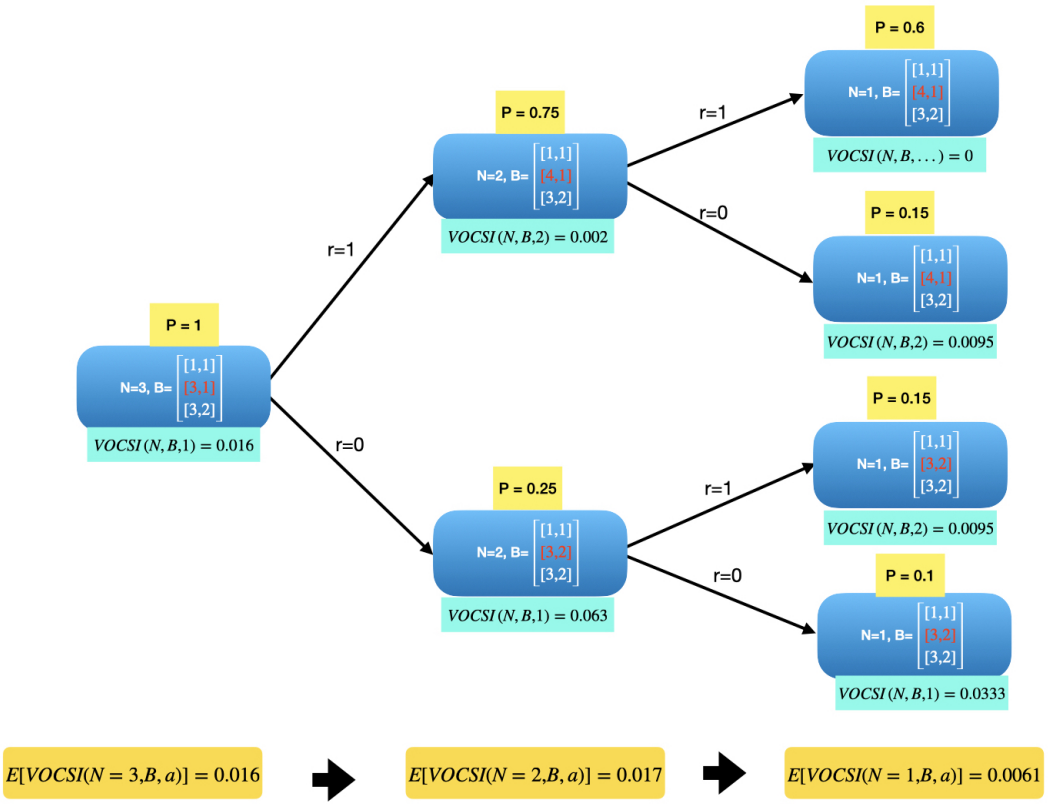

Figure 4.1: Example of the "When to ask for advice algorithm": The blue rectangles represent belief-augmented states, with the red arms indicating the actions chosen by the Bayes-optimal policy in each corresponding belief state. The light yellow rectangles denote the probabilities of reaching those states following the Bayes-optimal policy. The light blue rectangles represent the maximum VOCSI value associated with the adjacent belief state. The yellow rectangles indicate the expected VOCSI values for the respective horizon.

belief $b_0 = \begin{bmatrix}(1,1)\\(3,1)\\(3,2)\end{bmatrix}$ *assigned to each arm. The horizon of the problem-setting is set to* $T = 3$. *The expected VOCSI values can be observed in Figure 4.1 (Yellow rectangles in the bottom), indicating that waiting until the next step to ask for advice leads to a higher expected VOCSI value. The corresponding MDP solution for this example is depicted in Figure 4.2. We observe that if we immediately ask for advice regarding arm 1, the expected value* (q^*) *obtained would be* 2.2758. *However, if we first pull the arm and then seek advice in the subsequent step, the expected value obtained would be* 2.2772. *Thus, waiting to*

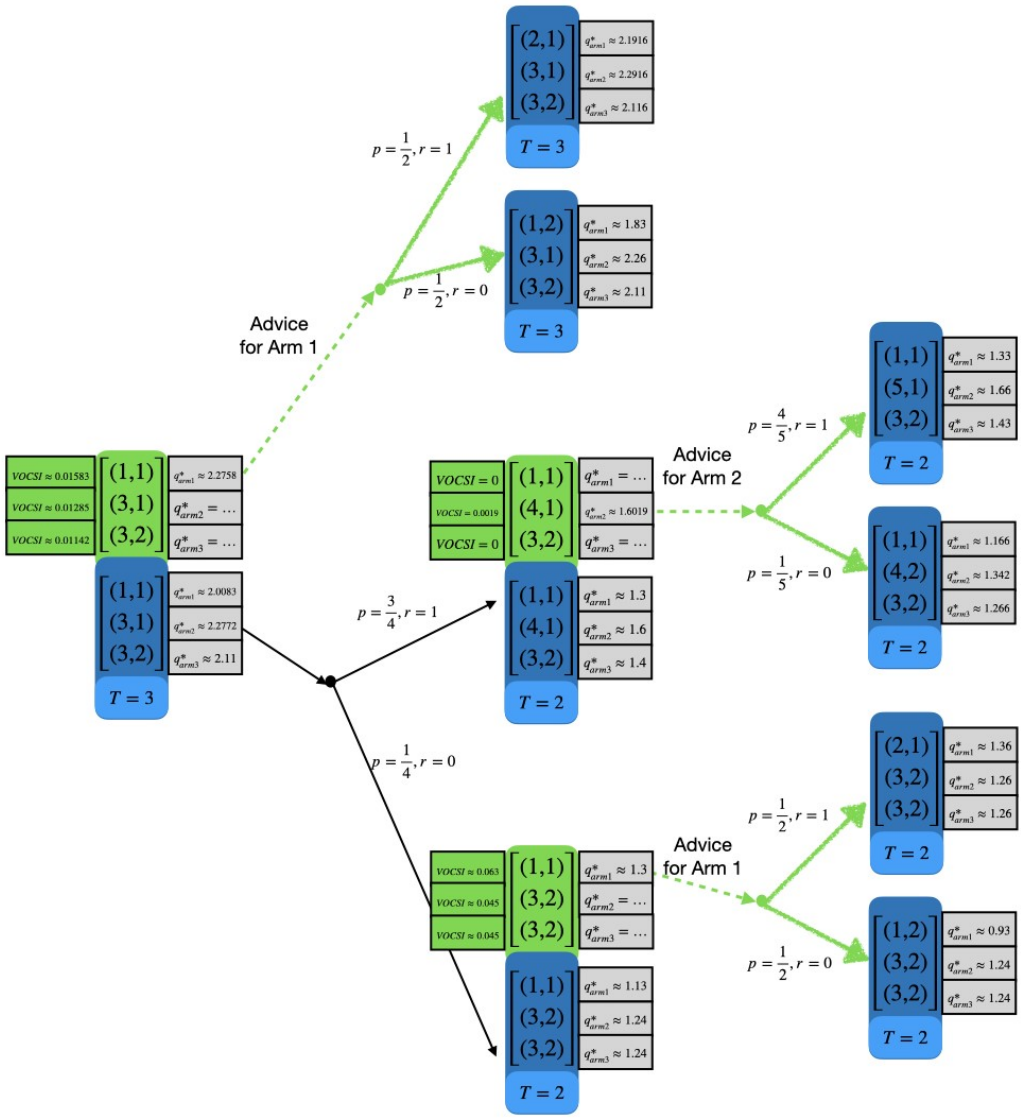

Figure 4.2: Example MDP of the "When to ask for advice problem": The green rectangles correspond to belief states in which the agent requests advice. The dark blue rectangles represent belief states in which exploration actions are taken, with the light blue rectangle indicating the horizon of the belief state. The green arrows depict actions involving advice-seeking, while the black arrows represent exploration actions. The gray rectangles represent the action values for the corresponding actions in this example MDP.

ask for advice results in a higher expected value, consistent with the findings presented in Figure 4.1.

4.2 Approximation of the Optimal Solution to "When and What to Ask about"

The optimal solution presented in Section 4.1 poses computational challenges. As the number of arms or the horizon in the problem-setting parameters increases, the algorithm becomes computationally intractable. Hence, it is necessary to develop an approximation for the optimal method.

One straightforward approach is to approximate the problem-setting horizon. We employ the same approximation technique introduced in Chapter 3, known as the "*h-myopic*" approach, where h represents the approximation horizon. When addressing the question of "When to ask for advice," if the true horizon of the problem-setting is denoted as T, we consider an approximation horizon of $h < T$. This approximation implies that instead of considering the expected gain values T time steps into the future, as in the case of the optimal solution, we only look ahead $h < T$ time steps.

4.3 Experiments

In this section, we present the experimental design (Section 4.3.1) and the subsequent analysis of the experiments (Section 4.3.2).

4.3.1 Design

We use an experiment design that investigates the "When to ask for advice problem" using either the optimal or approximation approaches. The "When to ask for advice problem" consists of three components: (1) "When should the agent ask for advice?", (2) "What arm should the agent ask about?", and (3) "Exploration policy," which is the Bayes-optimal policy in our case. These components are distinct because the agent needs to determine whether to postpone asking for advice or to inquire immediately ("When to ask for advice?"). Furthermore, if the agent decides to seek advice, it must determine which action to request advice for ("What arm to ask about?"). Finally, when advice is sought, the agent must follow an exploration policy to effectively

explore or exploit the environment ("Exploration policy"). Each of these three components can be approached using either the optimal or approximation solution. For this chapter, we focus on the advice part and use the Bayes-optimal policy as the exploration policy.

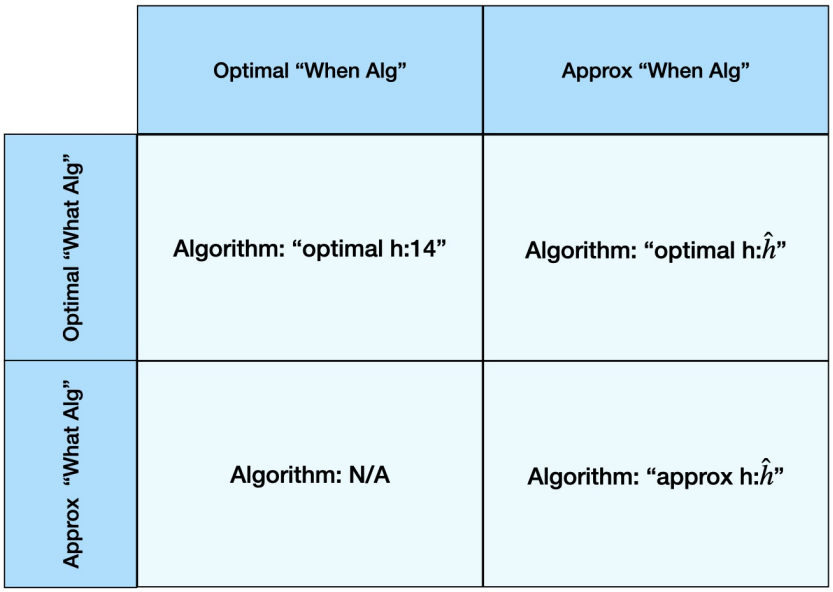

Figure 4.3: This figure presents four types of algorithm based on using the optimal solution or approximated solution for "When to ask for advice problem"and "What arm to ask about problem." The algorithm, in which we use approximate solution for "What arm to ask about problem" and the optimal solution for "When to ask for advice problem," is not available due to time-wise intractability.

Based on the considerations mentioned above, four distinct combinations arise between the approximation and optimal solutions for the "When to ask for advice problem"and the "What arm to ask about problem" that are illustrated in Figure 4.3. The details of these approximations can be found in Section 4.2. Out of these four combinations, we exclude the analysis of the combination involving the optimal solution for the "When to ask for advice problem"and the approximate solution for the "What arm to ask about problem." This omission is due to the intractability of this combination. Indeed, the combination involving the optimal solution for both the "When to ask for advice problem"and the "What arm to ask about problem" has the most computation among these four combinations, so some might ask "why it is not

intractable?" The reason behind this is that when using the optimal solution for the "When to ask for advice problem"and the approximate solution for the "What arm to ask about problem," we cannot share computations between the two problems since they have different horizons. However, when both problems are addressed using the optimal solution, most computations can be shared due to the identical horizon for both problems. Thus, we focus on the following three combinations:

- Optimal solution for the "When to ask for advice problem"with the optimal solution to the "What arm to ask about problem." The algorithm name used in this chapter for this solution is "optimal h:14", which indicates that we pick a horizon h=T=14 and calculate the optimal solution to "What arm to ask about problem."

- Approximate solution for the "When to ask for advice problem"with the optimal solution to the "What arm to ask about problem." The approximated horizon for the "When to ask for advice problem"is $\hat{h} \in \{1, \ldots, 14\}$. The algorithm name used in this chapter for this solution is "optimal h:$\hat{h}$" with $\hat{h}$ representing the value of the approximated horizon.

- Approximate solution for "When to ask for advice problem"with the approximate solution for "What arm to ask about problem." In this case, we use the same approximation horizon for both the "When to ask for advice problem"and the "What arm to ask about problem." The list of approximation horizons is defined as $\hat{h} \in \{1, \ldots, 6\}$. We limit the horizon range compared to the previous approximation to reduce computational time, as explained earlier in this section. The algorithm name used in this chapter for this solution is "approx h:$\hat{h}$" with $\hat{h}$ representing the value of the approximated horizon.

For our experiments, we use a Beta-Bernoulli Multi-armed bandit problem with three arms and a uniform prior belief to facilitate analysis. To ensure accuracy in the subsequent analysis (Section 4.3.2), we conduct the experiments

across a range of different initial beliefs (10^6 different beliefs). Specifically, we consider the following belief space:

$$B = \left\{ \begin{bmatrix} (\alpha_1, \beta_1) \\ (\alpha_2, \beta_2) \\ (\alpha_3, \beta_3) \end{bmatrix} \middle| \forall \alpha_i, \beta_i \in \{1, \ldots, 10\} \right\}$$

In the following sections, we analyze the "When to ask for advice algorithm" using various types of approximations outlined in Section 4.2. All results are averaged over the belief space before, consisting of 10^6 different initial beliefs. Due to the substantial number of initial beliefs ($N = 10^6$), error bars are omitted as they are negligible.

4.3.2 Analysis

In this section, we analyze the "When to ask for advice problem"using the optimal and approximation solutions discussed in Section 4.3.1. We employ two types of "When to ask for advice algorithm." The first approximation algorithm is referred to as "optimal h," which uses the optimal solution to calculate the VOCSI values and approximates the horizon of the "When to ask for advice problem"by using a specified value of h. For instance, "optimal h=5" considers a look-ahead horizon of $h = 5$ for the "When to ask for advice algorithm." Similarly, "optimal h=14" corresponds to the optimal solution of the "When to ask for advice problem" with a look-ahead horizon of $h = T = 14$. The second algorithm type is denoted as "approx h," which approximates both the look-ahead horizon of the "When to ask for advice problem"and the "What arm to ask about problem" (VOSCI) using the same value of h. For the "optimal h" algorithm, we conduct experiments for all possible values of $h \in 1, \ldots, 14$, but we only present a select few in the subsequent plots. Similarly, for the "approx h" algorithm, we conduct experiments for $h \in 1, \ldots, 6$ and only display a few chosen options in the analysis. The reason behind displaying only a few options is due to the readability of the figures.

In the subsequent sections, we analyze the "When to ask for advice problem"using both the optimal and approximation solutions just described. We explore various aspects to gain a comprehensive understanding of the problem.

Firstly, we investigate the optimal look-ahead horizon for the "When to ask for advice problem," providing valuable insights into the ideal look-ahead horizon size. Next, we examine the instances where we should delay the request for advice, allowing us to determine the number of states in which postponing the advice is beneficial. Furthermore, we assess the gains and losses incurred using the approximate "When to ask for advice problem". Lastly, we evaluate the performance measures of the proposed algorithms.

Optimal Look-Ahead Horizon for "When to ask for advice problem"

In this section, we analyze the minimum number of steps required to look ahead into the future to achieve the optimal solution for the "When to ask for advice problem." We use the optimal solution for the "What arm to ask about problem" derived in Chapter 3. Figure 4.4 illustrates that as we increase the true horizon from $T = 1$ to $T = 15$, the 98% confidence optimal look-ahead horizon gradually increases and converges around six steps. 98% confidence optimal look-ahead is the horizon that captures the optimal solution 98% of the times. Based on this finding, we can infer that, in our specific problem setting, the efficacy of advice in the later steps reaches its peak approximately six steps into the future. Consequently, if we need to postpone seeking advice, it is typically sufficient to delay it by a maximum of six steps on average to have optimal performance. Thus, in expectation, there is no need to extend the look-ahead horizon until the end to achieve a near-optimal outcome. We should mention that these results may vary for different belief spaces and extended horizons.

Wait Percentage

Figure 4.5 shows the percentage of belief states where it is beneficial to delay asking for advice. This means we could achieve a higher reward by seeking advice later rather than immediately. From the figure, it's evident that as the problem-setting horizon extends, more beliefs favor waiting to ask for advice. This suggests that advice becomes more valuable in longer horizons if we choose to delay the request. For horizons of $T = 1$ or $T = 2$, the wait

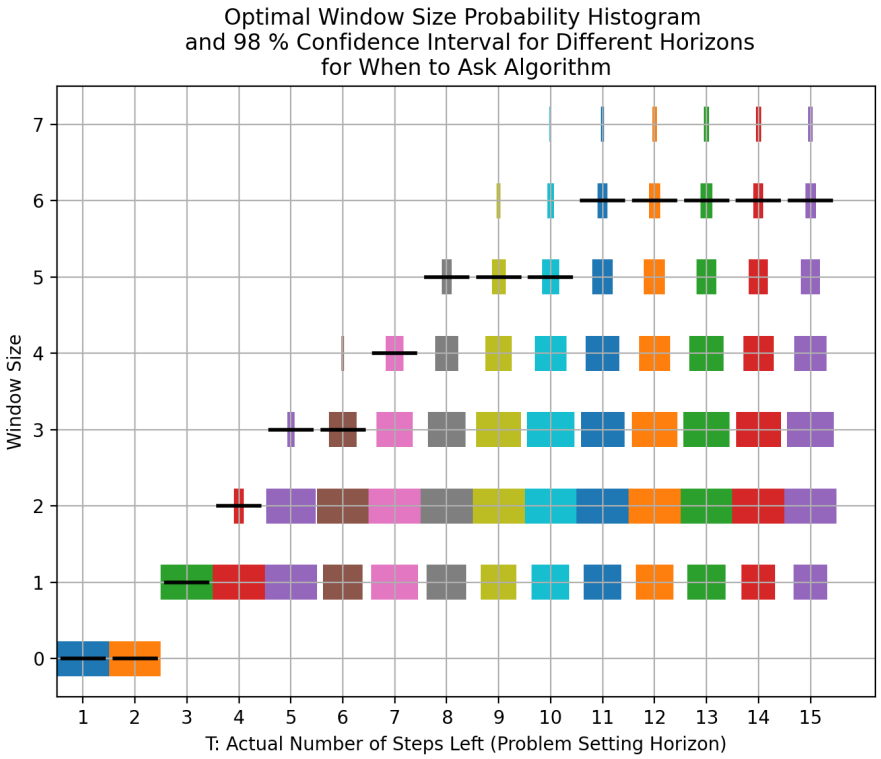

Figure 4.4: This figure presents a normalized histogram and the 98% confidence optimal window predictions when utilizing the optimal solution for the "what arm to ask about problem." The optimal window prediction represents the minimum number of steps we need to look ahead into the future to capture the optimal solution with 98% confidence for the "When to ask for advice problem." The black line represents the 98% confidence window size, while the remaining portion displays the normalized histogram of the optimal window sizes.

percentage is zero, meaning it is best to ask for advice immediately in these scenarios.

Gain and Loss of Delyaing Advice Using Approximate Solutions

In this section, we examine various measures related to gain, including maximum gain, maximum relative gain, average gain, and average relative gain. These measures provide insights into the effectiveness of different "When to ask for advice algorithms." You can find the detailed definition of these measures in the following.

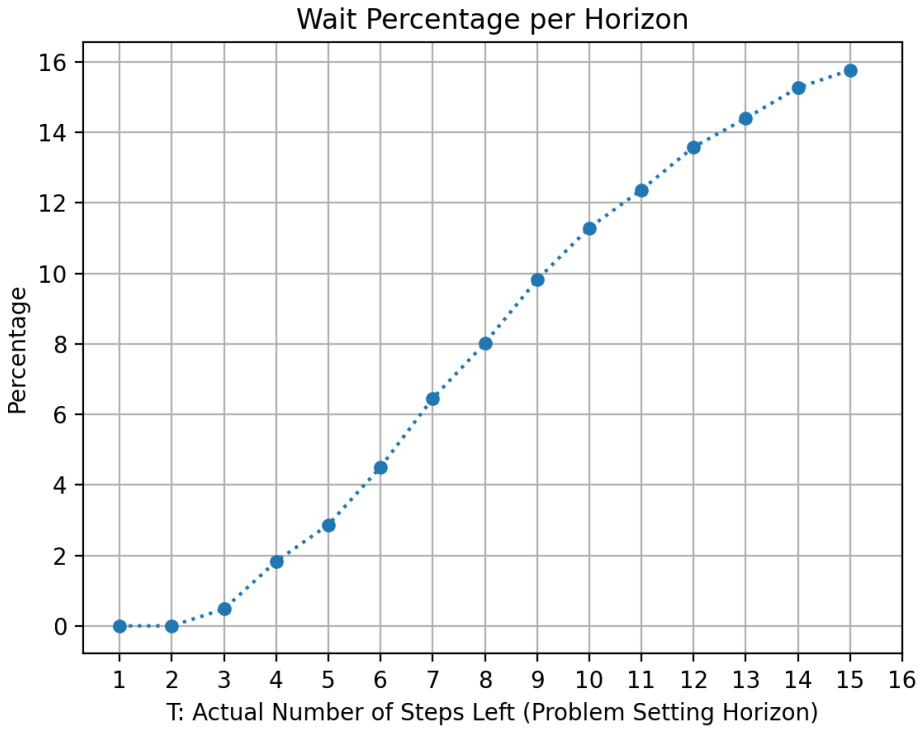

Figure 4.5: This figure illustrates the percentage of belief states in which postponing advice is more beneficial across different problem-setting horizons T.

- Gain: Represents the amount of gain achieved by following a specific "When to ask for advice algorithm" by asking for advice.
- Relative gain: Indicates the relative gain compared to the gain obtained by asking for advice immediately ($\frac{\text{approx algorithm gain} - \text{current step gain}}{\text{current step gain}}$).
- Max gain: Denotes the maximum possible gain attainable within the belief space.
- Max relative gain: Represents the maximum relative gain achievable within the belief space.
- Average loss: Quantifies the loss in gain incurred by following a particular "When to ask for advice algorithm" instead of the optimal solution.

- Average relative loss: Measures the relative loss in comparison to the gain achieved by the optimal solution ($\frac{\text{optimal solution gain}-\text{approx solution gain}}{\text{optimal solution gain}}$).

Based on the results from the previous Section (4.3.2), we observed an increasing number of beliefs that required the postponement of advice as the horizon increased. For the highest case with $T = 15$, the advice postponement was only necessary for sixteen percent of the belief space, indicating a relatively low percentage. Consequently, we decided to calculate the expectation of gain and relative gain results solely for positive prediction cases (instances where advice postponement is needed). Additionally, we computed the average loss results solely for false negative cases for the same reason.

Figure 4.6 demonstrates that as the approximation horizon h increases for the "optimal h" algorithm, the results for both gain and relative gain approach the optimal solution. Notably, even the optimal solution does not yield significant gains. This limited gain can be attributed to the nature of the advice, which consists of only one sample from the environment. Furthermore, the small gain indicates the characteristics of the three-armed Beta-Bernoulli bandit environment. These characteristics are such as a limited reward range and a few number of arms that could potentially prevent the effect of advice from taking place. Consequently, we introduced the measure of relative gain. The lower plot of Figure 4.6 reveals that the optimal solution yields an average relative gain of approximately four percent, indicating a four percent improvement if the "optimal h" algorithm with proper h is used. Unfortunately, due to computational constraints (time constraints), we were unable to compute the same results for the "approx h" algorithms.

Figures 4.7 and 4.8 depict the performance in terms of average loss and average relative loss, respectively, as the approximation horizon h increases for the "optimal h" algorithm. The plots reveal that our performance progressively approaches the optimal solution if we increase the approximation horizon h. Notably, the average losses for all algorithms are minimal, which can be attributed to the simplicity and limited impact of the advice in the decision-making process within the chosen small and simple environment. Moreover,

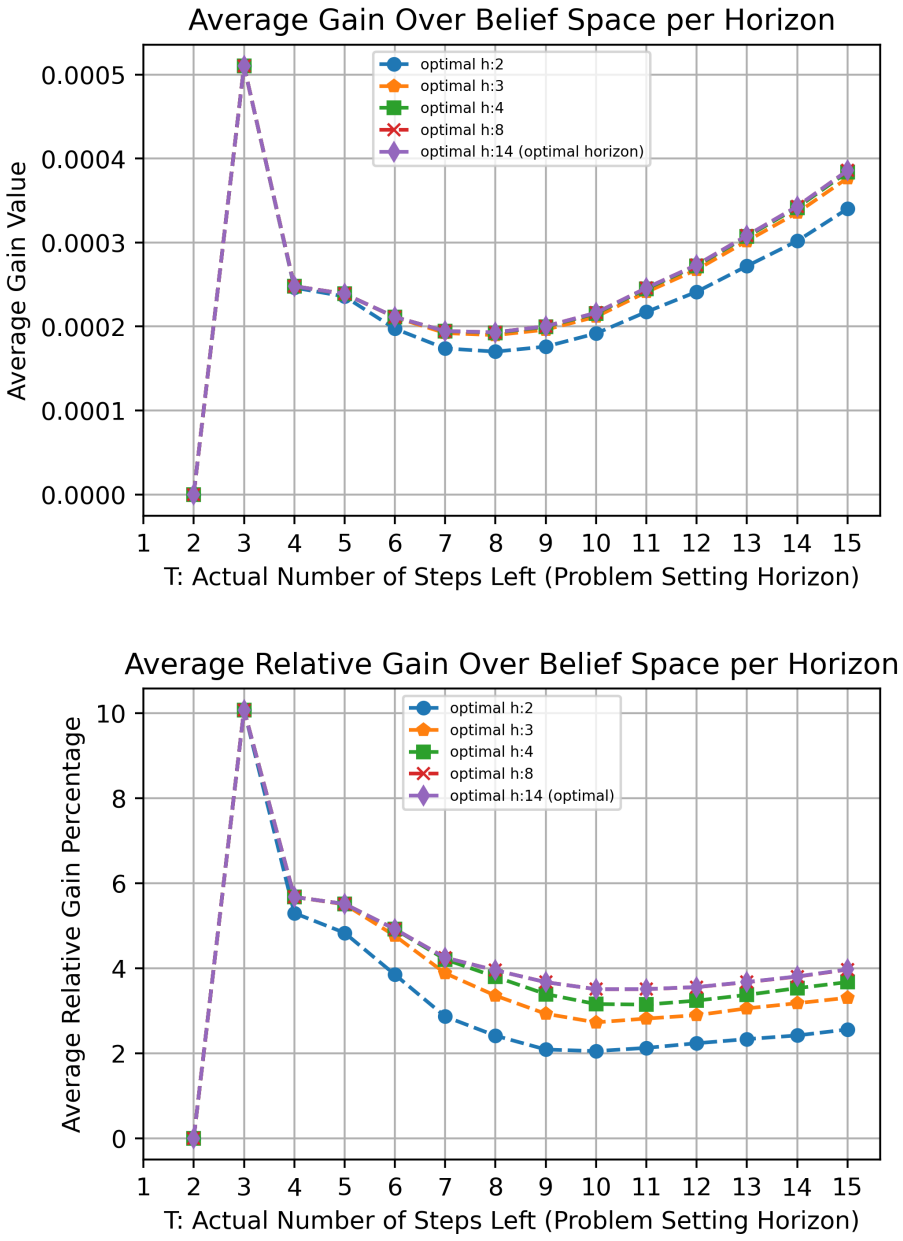

Figure 4.6: This figure displays the average gain and average relative gain for different algorithms across various problem-setting horizons T.

Figure 4.8 illustrates that the relative losses are also small, indicating near-optimal performance when the horizon is chosen such that $h \geq 3$.

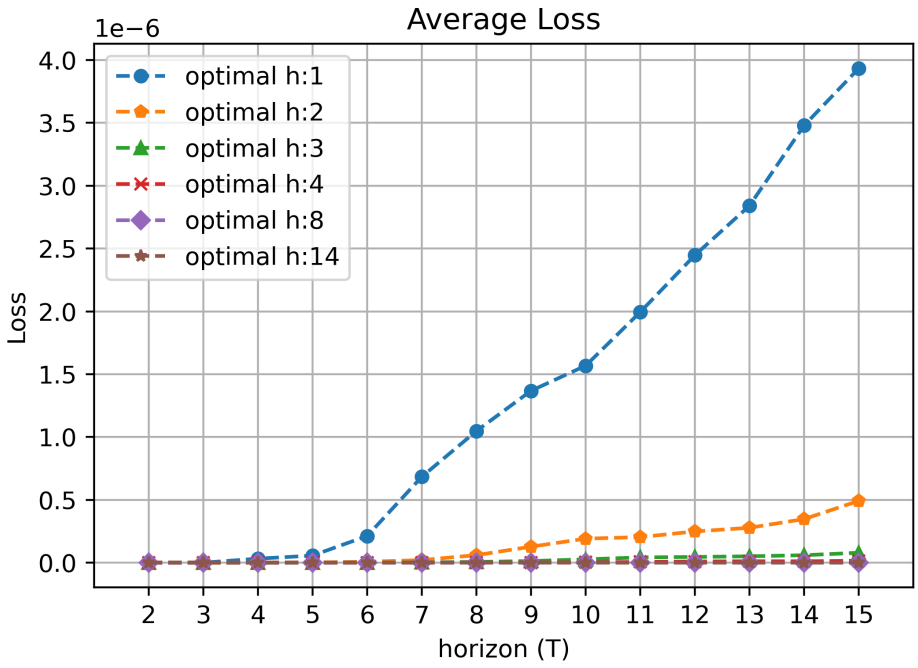

Figure 4.7: This figure presents the average loss for different algorithms across various problem-setting horizons T.

Performance

In this Section, we compare the performance of the optimal and approximate solutions (discussed in Section 4.3.1). We use fundamental performance measures such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates. The following are the definitions of these performance measures (see Figure 4.9):

- TP: True positive refers to the number of instances where the optimal prediction is positive (wait), and the approximate predictor also indicates a positive decision.
- TN: True negative represents the number of instances where the optimal prediction is negative (ask), and the approximate predictor suggests asking immediately.
- FN: False negative denotes the number of instances where the optimal

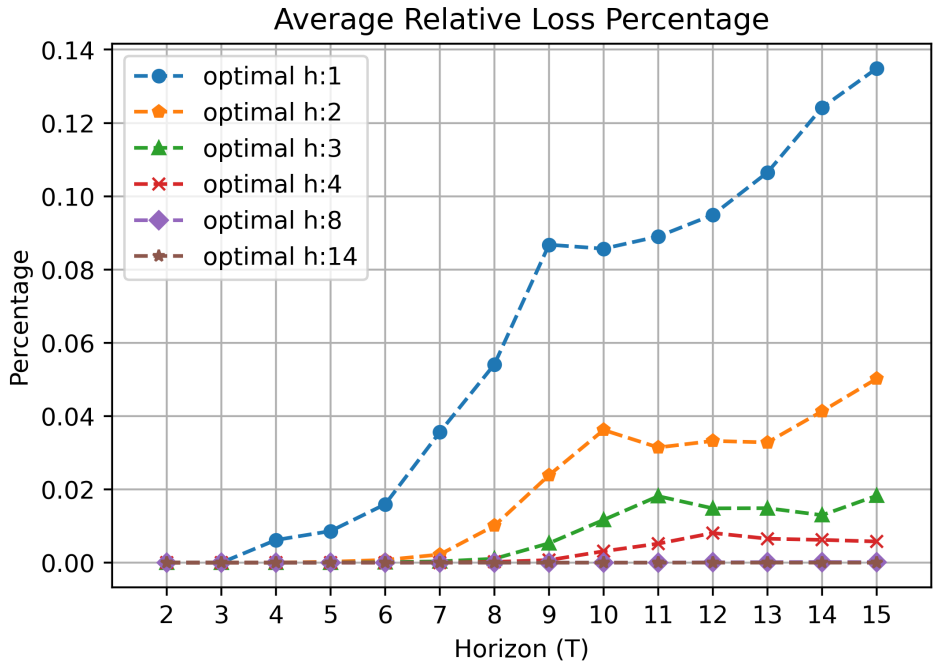

Figure 4.8: This figure illustrates the average relative loss for different algorithms across various problem-setting horizons T.

prediction is positive (wait), but the approximate predictor advises asking immediately.

- FP: False positive indicates the number of instances where the optimal prediction is negative (ask), but the approximate predictor suggests waiting.

Now, let us examine the results for the metrics mentioned above. Figure 4.10 illustrates that as we increase the approximation horizon for the "optimal h" algorithm, the TP results gradually converge to the optimal solution after $h = 4$. This pattern also applies to the "approx h" algorithm. However, the TN results are all the same as the optimal solution, regardless of the approximation horizon for the "optimal h" algorithm. This effect stems from the fact that if the optimal algorithm predicts a negative outcome (ask right away), it implies that no future step is expected to yield a higher gain than the current step. Therefore, the "optimal h" algorithm also predicts a negative

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

	$Y = A$	$Y = W$
$\hat{Y} = A$	**TN**	**FN**
$\hat{Y} = W$	**FP**	**TP**

Figure 4.9: Performance measures including true positive (TP), true negative (TN), false positive (FP), false negative (FN), precision, recall, and accuracy. The optimal solution is denoted by Y, and $\hat{Y}$ represents the algorithm prediction. The symbol A represents the decision to ask right away (negative prediction), while W represents the decision to wait (positive prediction).

outcome.

However, for the "approx h" algorithm, since we approximate the VOCSI, this argument does not hold. As shown in Figure 4.11, increasing the approximation horizon leads to a decrease in TN performance, which is unexpected. One possible reason for the poor performance of all "approx h" algorithms is that the average error (Figure 3.4) caused by the VOCSI approximation exceeds the average gain (Figure 4.6). Thus, a significant drop in performance is possible.

Further, in the analysis of FN and FP cases, Figure 4.12 demonstrates that as we increase the approximation horizon for the "optimal h" algorithm, the FN results gradually approach the optimal solution after $h = 4$. This trend also applies to the "approx h" algorithm. Additionally, Figure 4.13 indicates that there are no cases of FP for the optimal solution or the "optimal h" algorithm, regardless of the approximation horizon. This is because if the "optimal h" algorithm predicts a positive outcome, the optimal solution will also predict a positive outcome. However, the "approx h" algorithm exhibits

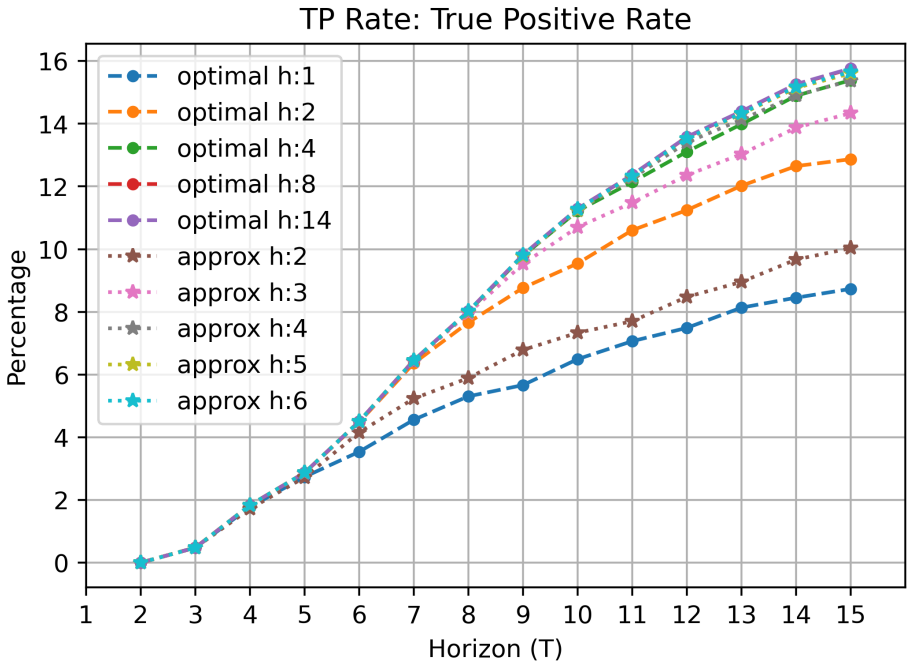

Figure 4.10: True Positive (TP) rate for different algorithms at varying problem-setting horizons (T).

poor performance, as depicted in Figure 4.13. This poor performance by the "approx h" algorithm is likely attributed to the average error (Figure 3.4) caused by the VOCSI approximation, which exceeds the average gain (Figure 4.6).

In addition to the basic performance measures mentioned above, there are more advanced performance measures, such as recall, accuracy, precision, and F1-score, which build upon the previous measures. The definitions of these measures are as follows (see Figure 4.9):

- Recall: Calculated as $Recall = \frac{TP}{TP+FN}$, this measure reflects the predictor's ability to identify cases where the optimal answer is positive (wait). In other words, it indicates the percentage of times we correctly identified the need to wait.

- Precision: Computed as $Precision = \frac{TP}{TP+FP}$, this measure assesses the predictor's accuracy in suggesting waiting (positive). It represents the

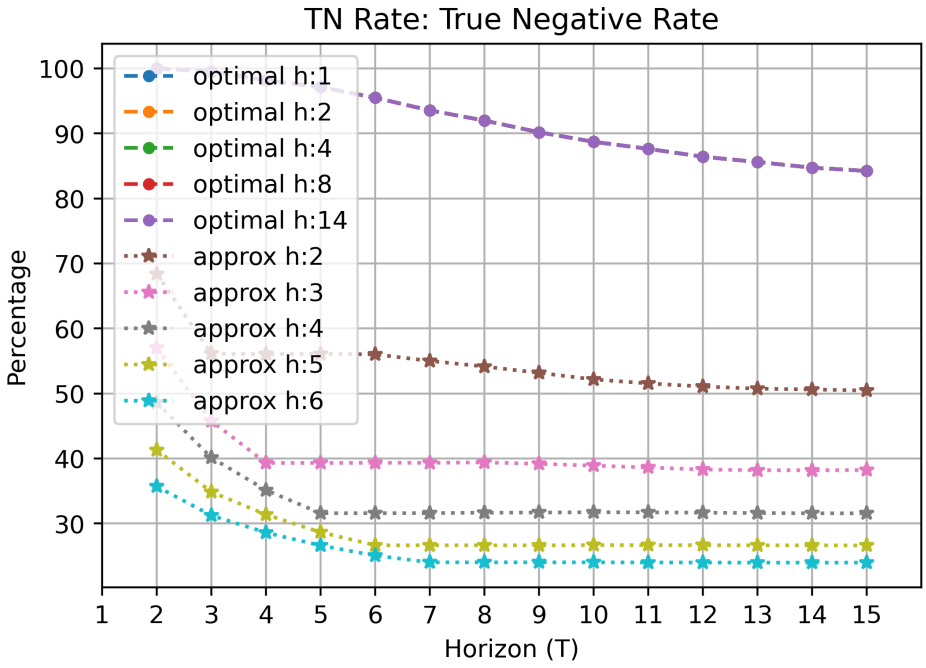

Figure 4.11: True Negative (TN) rate for different algorithms at varying problem-setting horizons (T).

percentage of times the predictor's wait recommendation was correct.

- Accuracy: Defined as $Accuracy = \frac{TP+TN}{TP+FP,TN+FN}$, this measure indicates the proportion of correct predictions out of all predictions made.

- F1-score: Calculated as $F1, score = 2 \times \frac{precision*recall}{precision+recall}$, the F1-score is the harmonic mean of precision and recall. It provides a balanced metric that takes into account both precision and recall. As with other metrics, a higher value indicates better performance. The use of the harmonic mean is justified by the fact that both precision and recall are rates, making a normal mean inappropriate.

Now, let us examine the results for these metrics. Figure 4.14 demonstrates that as we increase the approximation horizon for the "optimal h" algorithm, the recall results gradually approach the optimal solution after $h = 4$. This trend is also observed for the "approx h" algorithm. Regarding precision, the "optimal h" algorithm consistently achieves 100% precision. Since we employ

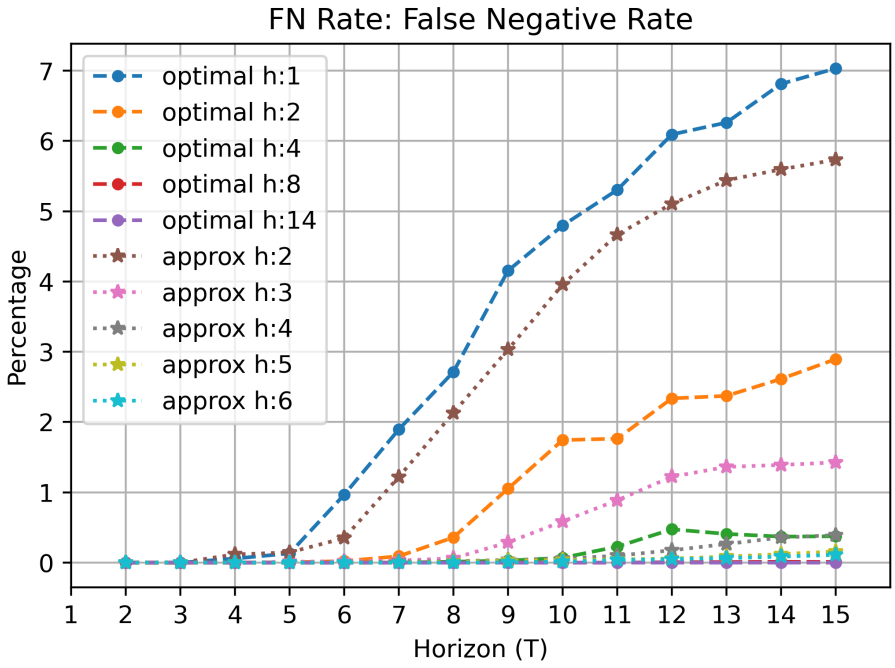

Figure 4.12: False Negative (FN) rate for different algorithms at varying problem-setting horizons (T).

the optimal solution for VOCSI, the positive predictions of this algorithm form a subset of the optimal predictions. Consequently, if the "optimal h:$\hat{h}$" $\forall \hat{h} \in \{1, \ldots, 14\}$ algorithm predicts a positive outcome, the ground truth "optimal h:14" will also be positive, eliminating the need for plotting. However, the same reasoning does not hold for the "approx h" algorithm as we do not use the optimal solution for VOCSI ("What arm to ask about problem"). As shown in Figure 4.15, the precision for each "approx h" algorithm starts from zero percent and increases as the horizon increases but only reaches around twenty percent. The poor precision is primarily attributed to the high false positive (FP) rate exhibited by the "approx h" algorithm (Figure 4.13).

Turning to accuracy, Figure 4.16 illustrates that the accuracy of the "optimal h" algorithm approaches an approximately optimal level as the approximation horizon increases. Moreover, it consistently performs near-optimal for all horizons. In contrast, as seen in Figure 4.16, the accuracy of each "approx h" algorithm is consistently far from optimal. Furthermore, the accuracy

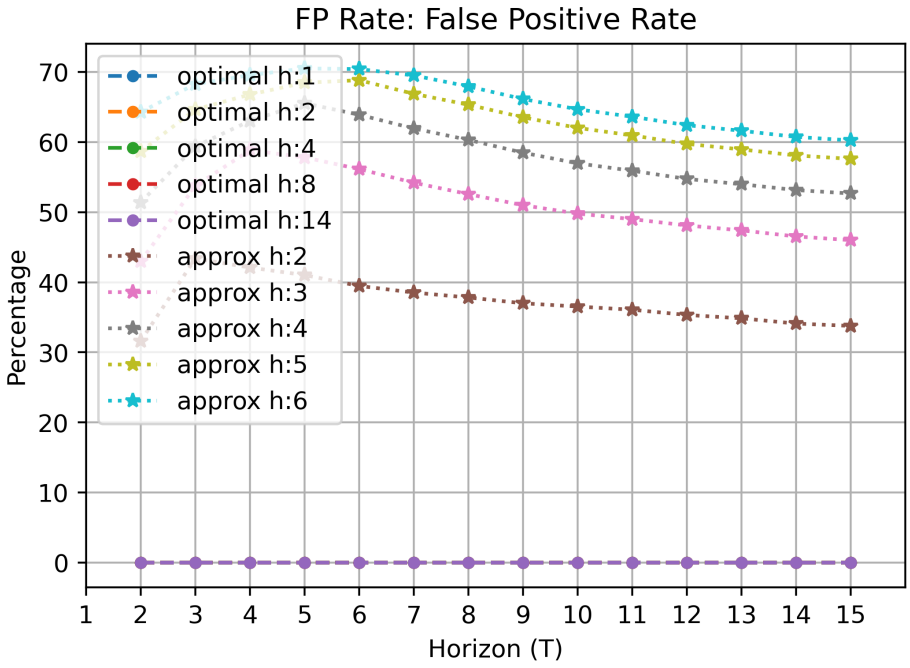

Figure 4.13: False Positive (FP) rate for different algorithms at varying problem-setting horizons (T).

drops as the approximation horizon increases, which is unexpected. One possible explanation for this behavior is that higher horizons increase the chances of making errors.

Examining the F1-score, Figure 4.17 reveals that the F1-score of the "optimal h" algorithm approaches the optimal level as the approximation horizon increases, becoming near-optimal for $h \geq 4$. However, for the "approx h" algorithm, as depicted in Figure 4.17, the F1-score for each "approx h" algorithm is consistently far from optimal.

Based on the results and plots from this section, we can conclude that in our setting, we can use the "optimal h" algorithm with $h \geq 4$ since it closely approximates the optimal solution. Unfortunately, the other double approximation technique ("approx h" algorithms) does not yield satisfactory results. We suspect that the poor performance of the "approx h" algorithms stems from errors introduced by the approximation of the VOCSI ("What arm to ask about problem").

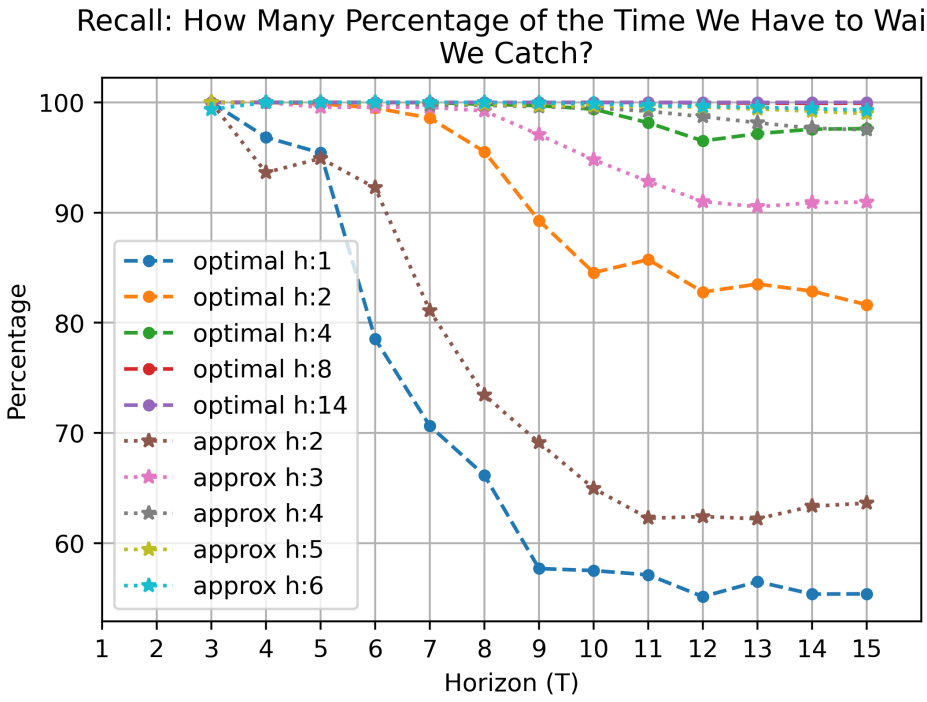

Figure 4.14: Recall percentage for different algorithms at varying problem-setting horizons (T).

4.4 Conclusion

This chapter addressed the "when to ask for advice problem" by proposing the optimal solution and introducing two approximation algorithms: the "optimal h" algorithm and the "approx h" algorithm. Through our analysis, we made several observations.

We discovered that for the "when to ask for advice problem," an average look-ahead window size of $h = 6$ is sufficient while using the optimal solution for the 'what to ask for advice problem" to achieve optimal solution with 98% confidence. This means delaying advice-seeking by up to six steps on average is almost as effective as extending the look-ahead horizon to the end. Additionally, we discovered that as the problem-setting horizon extends, the significance of advice grows, making it more beneficial to delay advice-seeking than to seek advice immediately.

Our analysis of the "optimal h" algorithm reveals that this algorithm pro-

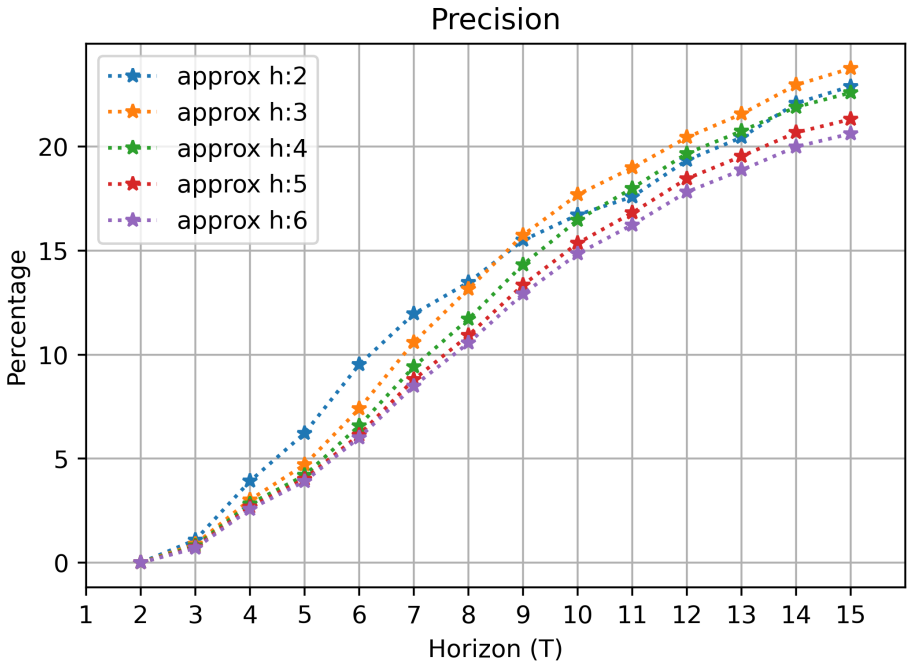

Figure 4.15: Precision percentage for different algorithms at varying problem-setting horizons (T).

gressively approximates the optimal solution as the approximation horizon h increases. Both the gain and relative gain show convergence to this optimal solution. Similarly, increasing the approximation horizon improves the average loss and average relative loss, bringing the performance closer to optimality. Even though the nature of advice in our context might cap the gains from delaying advice, the relative gain remains notably high in our presented problem-setting.

Moreover, we found that the "optimal h" algorithm exhibits superior performance compared to the "approx h" algorithm in terms of TP, TN, FP, and FN. The recall, precision, accuracy, and F1-score analysis further confirm the superiority of the "optimal h" algorithm over the "approx h" algorithm. On the other hand, the "approx h" algorithm struggles due to VOCSI approximation errors and fails to achieve satisfactory results. This analysis strongly suggests that the "optimal h" algorithm offers a much closer approximation to the optimal solution, deeming it a more suitable choice.

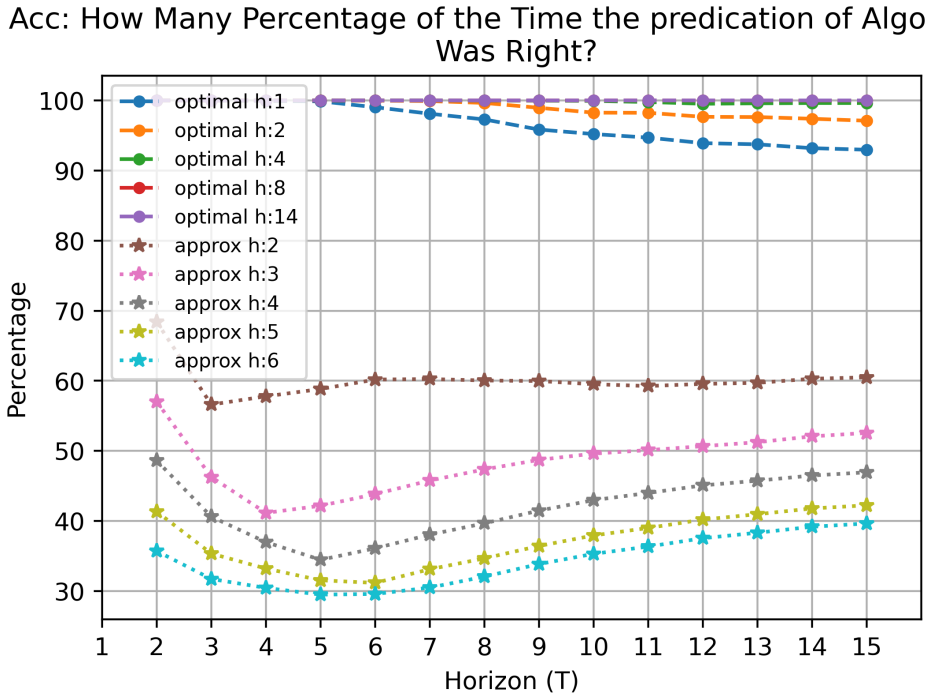

Figure 4.16: Accuracy percentage for different algorithms at varying problem-setting horizons (T).

In conclusion, the findings from this chapter highlight the effectiveness of the "optimal h" algorithm and emphasize the limitations of the alternative approximation method. The poor performance of the "approx h" algorithm can be attributed to its approximation of the "what to ask for advice problem." Therefore, we recommend utilizing the "optimal h" algorithm for solving the "what to ask for advice problem," considering its favorable performance and closer approximation to the optimal solution.

w

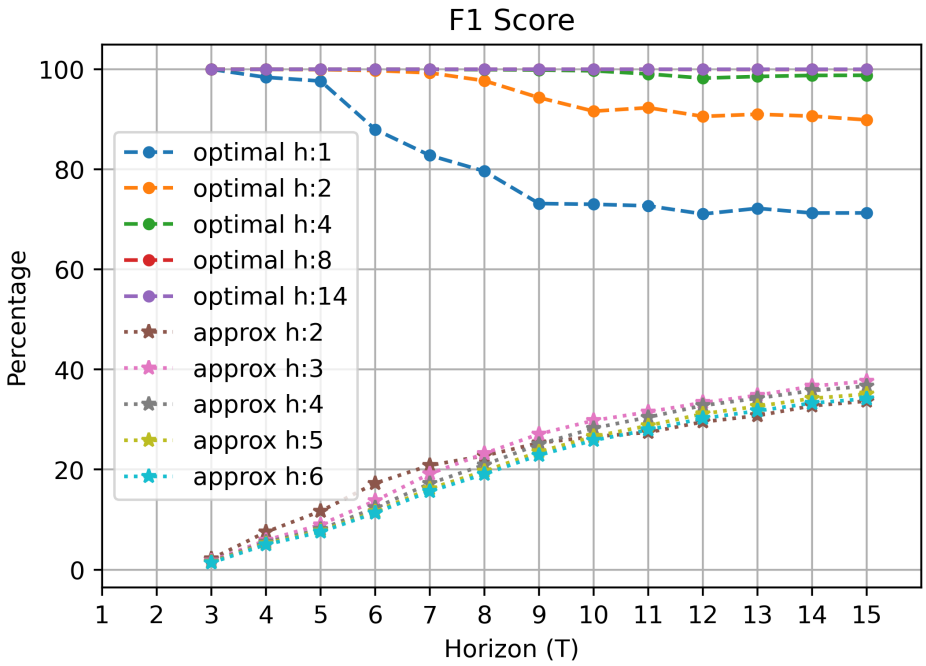

Figure 4.17: F1-score percentage for different algorithms at varying problem-setting horizons (T).

Chapter 5

Optimal Exploration

In this chapter, we discuss the relationship between the Bayesian Q-learning algorithm, as outlined in Section 2.6.5, and the Bayes-optimal policy discussed in Section 5.1. Then, we provide an analysis and development of a specific type of approximation for the Bayes-optimal and finite-horizon Gittins' policy. This approximation, which we explore in Sections 5.2 and 5.3, is centered around approximating the problem setting horizon.

Our contribution lies in analyzing an approach to managing the complexity inherent in Bayesian methods for optimal exploration, particularly in environments with a finite horizon. By approximating the problem setting horizon, we offer a practical solution to apply Bayes-optimal policies in a more computationally tractable way.

The significance of this work extends to a wide range of applications in reinforcement learning and decision-making processes, where understanding and balancing exploration and exploitation are crucial. Our approach provides insights into achieving near-optimal exploration strategies, potentially enhancing the efficiency and effectiveness of decision-making algorithms in uncertain environments.

5.1 Bayesian Q-learning vs. Bayes-optimal Policy

In this section, we explore the connection between Bayesian Q-learning (Section 2.6.5) and the Bayes-optimal policy. The Bayesian Q-learning policy is

defined as $\pi = \arg\max_a \left\{ \mathop{\mathbb{E}}_{b_a}[X^a] + \mathit{VPI}(b, a) \right\}$.

Claim 6. *For a given belief b, the Bayesian Q-learning policy mentioned above is equivalent to the Bayes-optimal policy with a horizon of two:*

$$\pi = \arg\max_a q^*(w = (n = 2, b), a))$$

Proof. Based on Claim 4, we have $\mathit{VPI}(b, a) = \mathit{1\text{-}myopic\ VOCSI}(b, a)$. Additionally, according to Claim 2, we know that $\arg\max_a q^*(w = (n = 2, b), a)) = \arg\max_a (\mathop{\mathbb{E}}_b[X^a] + \mathit{VOCSI}(1, b, a))$. By combining these equations with the Bayesian Q-learning policy, we observe that the Bayesian Q-learning policy is equivalent to the Bayes-optimal policy with a horizon equal to two ($\pi = \arg\max_a q^*(w = (n = 2, b), a))$). □

Based on Claim 6, it is evident that the Bayesian Q-learning policy is not optimal when the problem setting's horizon is not two ($n \neq 2$). Furthermore, we can understand the rationale behind the naming of the 1-myopic VOCSI, as it represents a VOCSI with a fixed horizon of one, regardless of the actual horizon, thus justifying the term "myopic."

5.2 Approximate Optimal Exploration

The optimal policy that achieves the global minimum of Bayesian regret is the Bayes-optimal policy (Section 2.6.3). However, this method becomes intractable due to the exponentially large state space of the BAMDP when dealing with a large number of actions and large horizons, even in the MAB setting. Gittins proposed an algorithm that aims to expedite the computation of the Bayes-optimal policy by reducing the problem to an index policy (Section 2.6.4). Nonetheless, this approach remains computationally infeasible for very large horizon values. Consequently, there is a need for approximations to reduce these computational challenges.

In the remainder of this chapter, we investigate and analyze a specific type of approximation that involves using a myopic horizon instead of the actual one by truncating the horizon. We apply this technique to both the Bayes-optimal policy and the Gittins' policy using two types of horizon truncation:

constant and dynamic. These techniques will be explained in detail in the subsequent sections.

5.2.1 H-myopic Optimal

As mentioned earlier, the optimal policy is not tractable. Therefore, in this section, we choose a truncated horizon approach, employing two horizon truncation techniques:

- Constant horizon: In this technique, we select a fixed horizon n' and maintain that horizon for our policy, irrespective of the number of steps remaining.
- Dynamic horizon: In this technique, we choose an initial horizon n', and as we progress at each time step, we decrease the horizon value. If the horizon value reaches one, we keep it at that value for subsequent steps. This technique is more aligned with the horizon selection procedure of the Bayes-optimal policy, which tends to select the actual horizon as the time step increases. For instance, if the horizon is n' at time t, it will be $n' - 1$ at the next time step $t + 1$.

5.2.2 H-myopic FH-Gittins

As previously mentioned, while Gittins' index policy offers improved computational efficiency compared to the Bayes-optimal policy, it still encounters challenges when dealing with large horizons. Therefore, in this section, we adopt the same constant and dynamic horizon truncation techniques just described (Section 5.2.1) and apply them to the FH-Gittins policy.

5.3 Experiments

In the next two sections, we will evaluate the performance of our approximate algorithms. First, we present the experiment designs of this evaluation (Section 5.3.1) based on different horizon types for the methods discussed in Section 5.2. We also discuss the baselines (Section 5.3.2) used in the experiments, which

include several methods derived for the Bayesian learning setting. Finally, we present the results in Section 5.3.3.

5.3.1 Design

In this section, we describe the experiment design involving various horizon values to overestimate or underestimate the horizon for the approximate methods introduced in Sections 5.2.1 and 5.2.2. The following conditions are considered for the estimated horizon values:

- Underestimated Horizon for h-myopic optimal method
- Overestimated Horizon for h-myopic optimal method
- Underestimated Horizon for h-myopic FH-Gittins method
- Overestimated Horizon for h-Myopic FH-Gittins method

Now, let us discuss the experiment design. Our problem setting involves a finite-horizon Bayesian three-armed Beta-Bernoulli multi-armed bandit. For each method (h-myopic optimal or h-myopic FH-Gittins) designed for a specific horizon, we use a uniform prior belief b_0. The actual horizon is set to $T = 50$, and we run each experiment for $N = 10^5$ iterations. In each independent iteration, the true parameter of the multi-armed bandit is drawn from the prior belief b_0.

Additionally, for the h-myopic optimal methods, we consider a set of horizons $h \in \{1, 2, 4, 8\}$.

5.3.2 Baselines

The baseline algorithms against which we compare our results are as follows:

- Bayes-optimal policy (optimal): This method aims to minimize the Bayes regret and serves as the optimal baseline. Since our work is based on the Bayesian learning setting with the Bayes regret as the objective, this method provides an optimal benchmark.

- Thompson sampling (TS): Thompson sampling is one of the earliest algorithms designed for the Bayesian learning setting. It is computationally straightforward and serves as an important baseline in our experiments since it is asymptotically optimal.

- Upper confidence bound (UCB): UCB is a fundamental policy for minimizing frequentist regret. Comparing our proposed methods with UCB provides valuable insights and serves as another baseline for evaluation.

- Bayesian Q-learning method (2-myopic optimal): This method is derived from the Bayesian Q-learning paper [6]. As mentioned in Section 5.1, this method is equivalent to the h-myopic optimal approximation method with $h = 2$. Therefore, it is automatically included in the results and referred to as *h-myopic* $= 2$ in the legends of some figures.

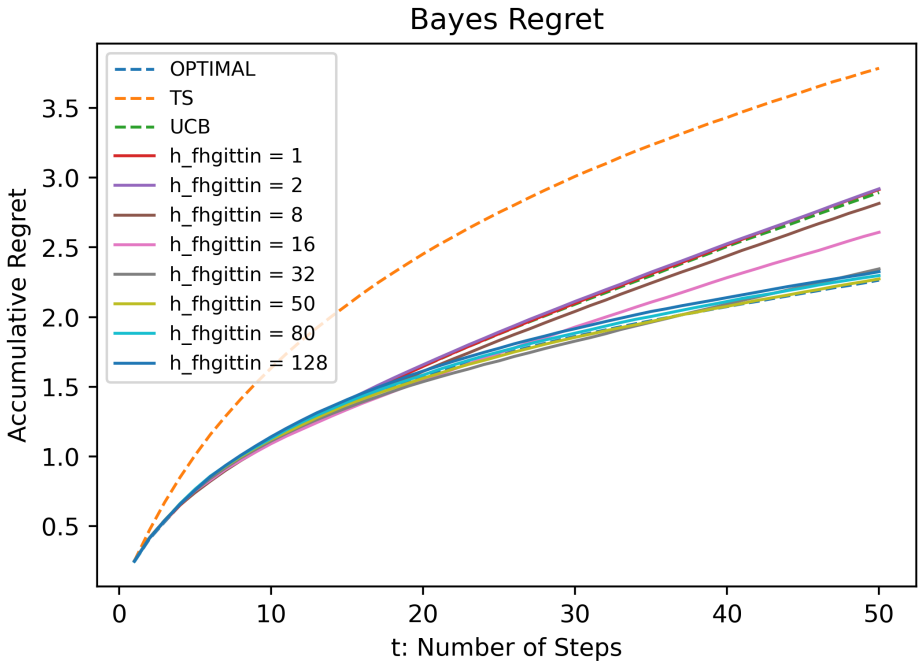

Figure 5.1: Bayes regret of the dynamic horizon h-FH-Gittins policy compared to the UCB, TS, and Bayes-optimal policies, with a true horizon of $T = 50$. For example, the parameter "h-fhgittin=32" represents the dynamic horizon h-FH-Gittins policy with a horizon of 32.

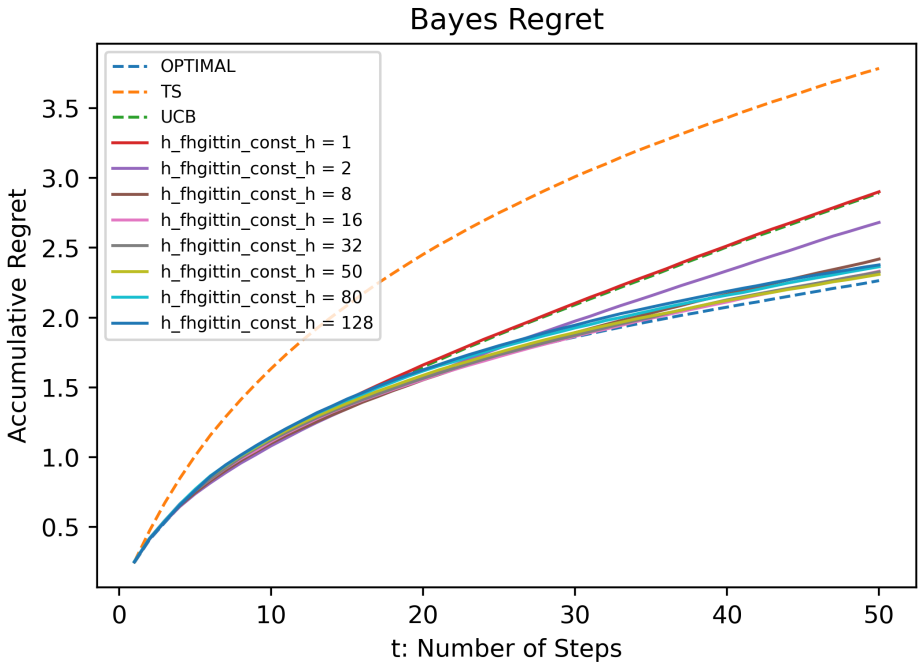

Figure 5.2: Bayes regret of the constant horizon h-FH-Gittins policy compared to the UCB, TS, and Bayes-optimal policies, with a true horizon of $T = 50$. For example, the parameter "h-fhgittin-const-h=32" represents the constant horizon h-FH-Gittins policy with a horizon of 32.

5.3.3 Results

In this section, we present the results and analyze the figures. All the graphs in this section display the Bayes regret for different methods when the experiment's horizon is $T = 50$. The results are averaged over 10^5 samples from the prior belief b_0 for the environments. For instance, $BR_{b_0}(t, \mathcal{A})$ represents the cumulative Bayes regret for method $\mathcal{A}$ up to time t with a true horizon of $T = 50$. Due to the large number of samples ($N = 10^5$), the error bars are negligible and thus not included in the figures.

Figures 5.1, 5.2, 5.3, and 5.4 illustrate the Bayes regret for different approximations of the Bayes-optimal policy with varying horizons (dynamic horizon h-FH-Gittins, constant horizon h-FH-Gittins, dynamic horizon h-myopic optimal, and constant horizon h-myopic optimal, respectively). The results are compared to the Bayes-optimal policy, UCB, and Thompson sampling (TS).

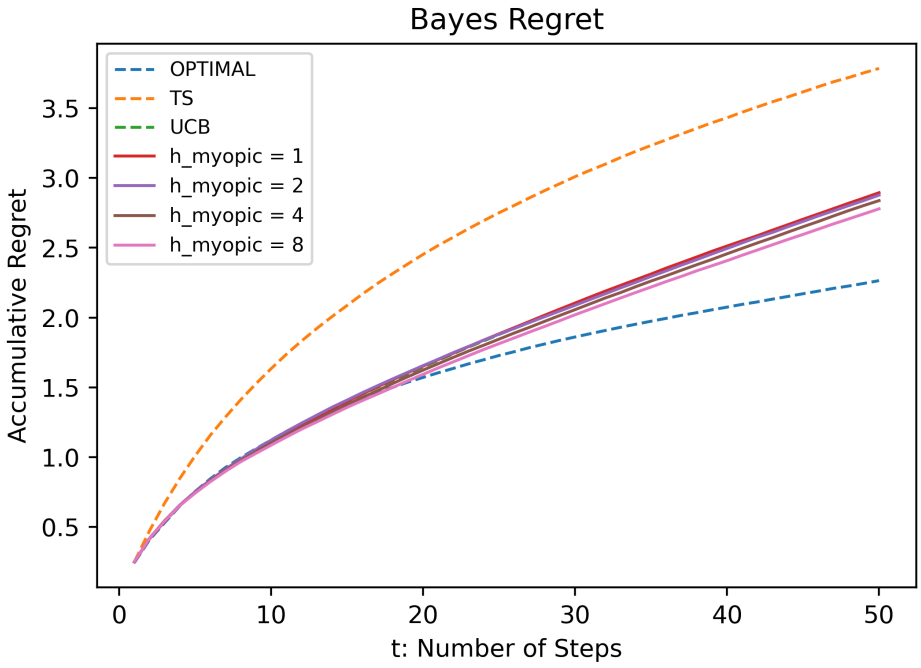

Figure 5.3: Bayes regret of the dynamic horizon h-myopic optimal policy compared to the UCB, TS, and Bayes-optimal policies, with a true horizon of $T = 50$. For example, the parameter "h-myopic=4" represents the dynamic horizon h-myopic optimal policy with a horizon of 4.

As observed from these graphs, increasing the horizon leads to a closer performance (lower Bayes regret) to that of the Bayes-optimal policy. **Importantly, even when overestimating the horizon ($h > T$), the performance is significantly better than underestimating it.**

These results raise questions such as "Why do we need both constant horizon and dynamic horizon?" and "Which one is better?" To address these questions, we compare the dynamic and constant horizon variations for the h-FH-Gittins policy in a pairwise manner (Figure 5.5). **Remarkably, the constant horizon h-FH-Gittins algorithm consistently outperforms the dynamic horizon h-FH-Gittins algorithm for all horizons. However, as the approximation horizon h increases, the performance gap diminishes.** A contributing factor to this trend is illustrated by considering an approximation horizon $h = 8$ alongside a true horizon $T = 50$. After progressing through 8 steps in the dynamic approach, the approximation horizon

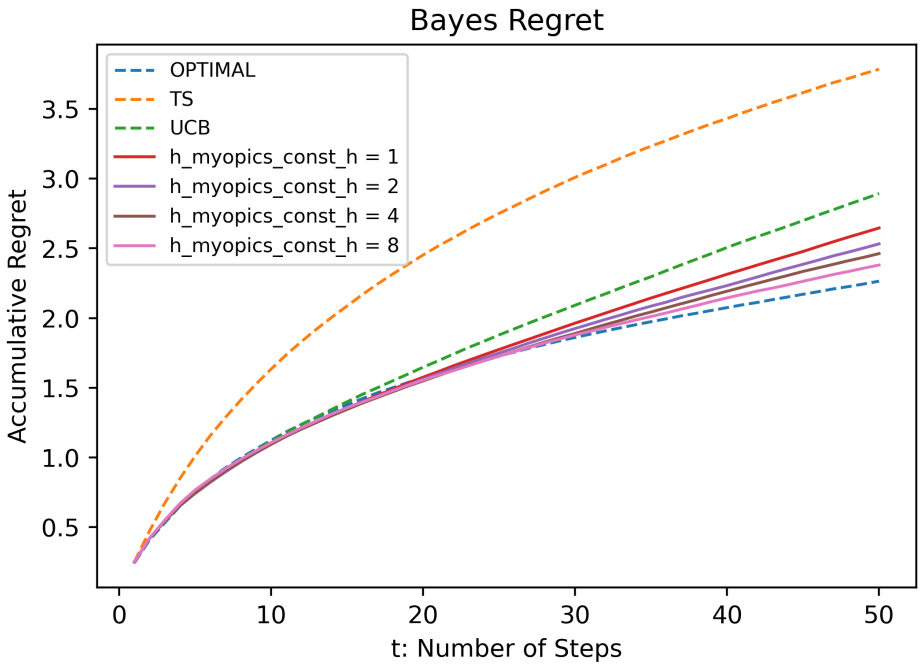

Figure 5.4: Bayes regret of the constant horizon h-myopic optimal policy compared to the UCB, TS, and Bayes-optimal policies, with a true horizon of $T = 50$. For example, the parameter "h-myopic-const-h=4" represents the constant horizon h-myopic optimal policy with a horizon of 4.

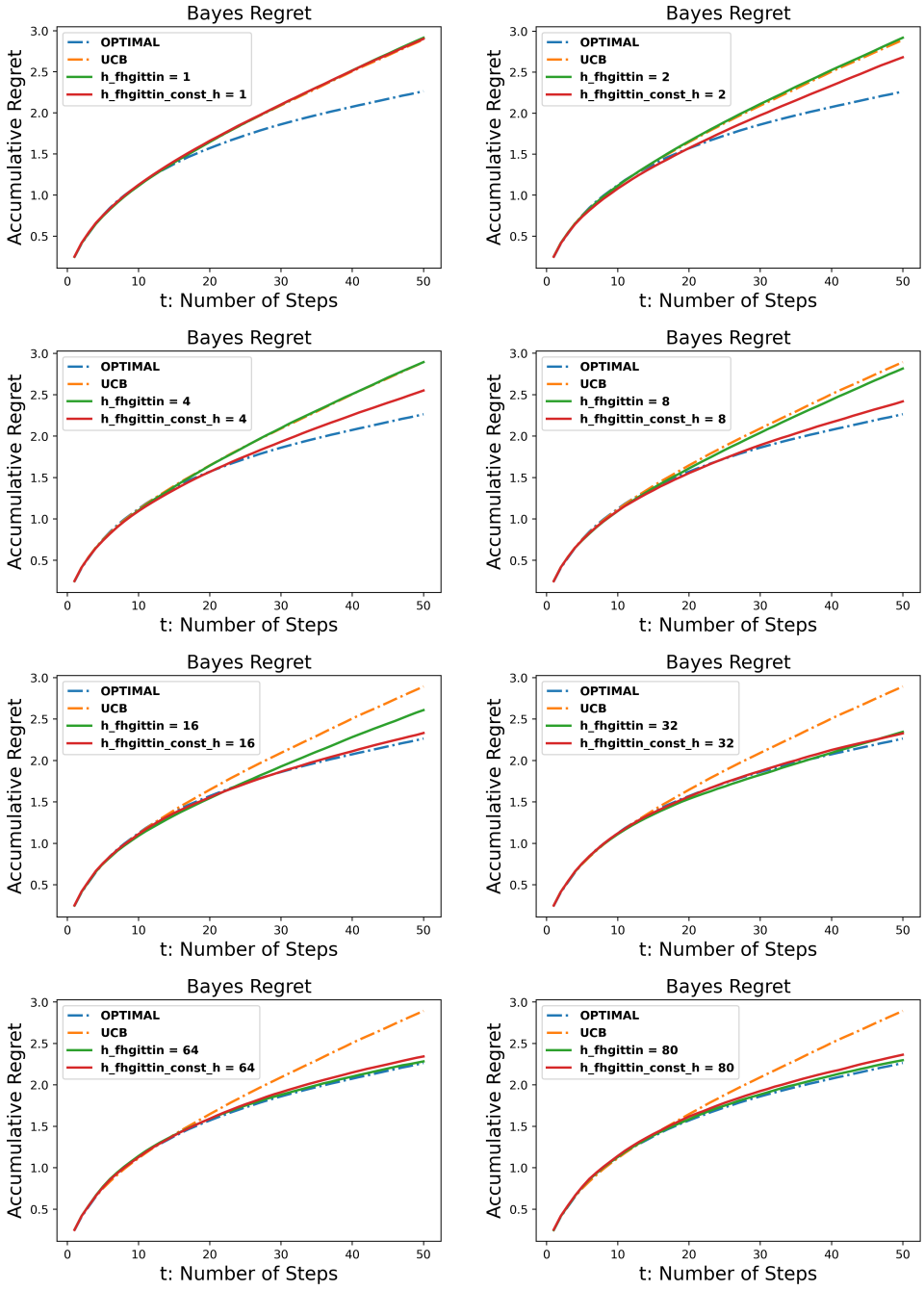

Figure 5.5: Comparison of Bayes regret between the constant horizon h-FH-Gittins and dynamic horizon h-FH-Gittins strategies for different pairs of myopic horizons, with a true horizon of $T = 50$. For example, the parameter "h-fhgittin=32" represents the dynamic horizon h-FH-Gittins policy with a horizon of 32.

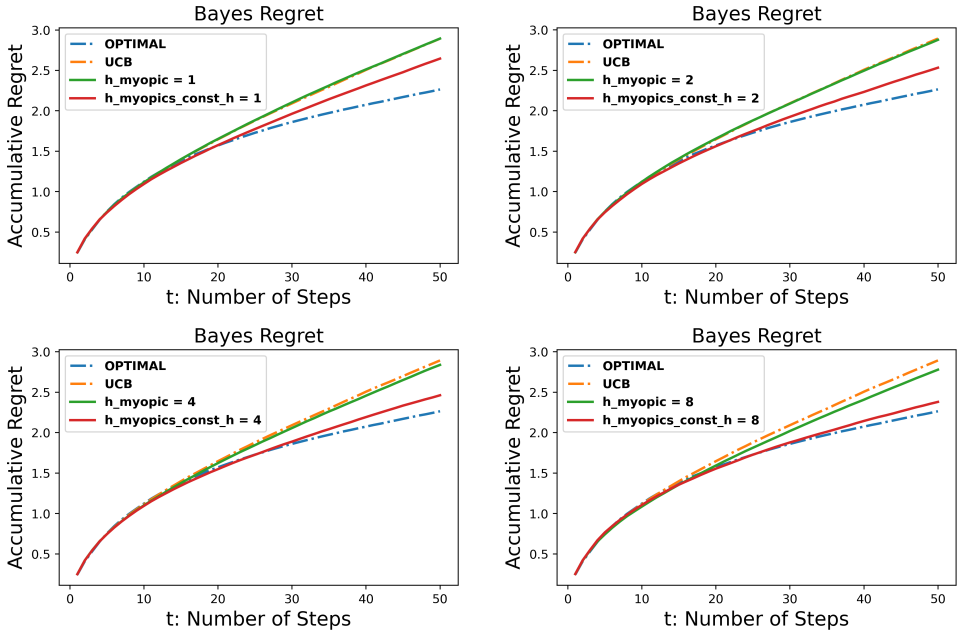

Figure 5.6: Comparison of Bayes regret between the constant horizon h-Myopic Optimal and dynamic horizon h-Myopic Optimal strategies for different pairs of myopic horizons, with a true horizon of $T = 50$. For example, the parameter "h-myopic=4" represents the dynamic horizon h-myopic optimal policy with a horizon of 4.

effectively reduces to $h = 1$ for the remainder of the experiment. On the other hand, the constant horizon approach provides predictions consistently for eight future steps, which is superior to the one-step prediction in the dynamic horizon approach (5.2). **Similar results and behavior are observed for the h-myopic optimal policy, as depicted in Figure 5.6.**

Having established that the constant horizon is a better choice for horizon approximation in the experimental setting, we compare the h-FH-Gittins and h-myopic optimal policies with the constant horizon (Figure 5.7). **It is evident that a higher horizon must be chosen for h-FH-Gittins to achieve the same level of performance as h-myopic optimal. However, this trade-off is justified by the fact that h-FH-Gittins is much faster than h-myopic optimal (Figure 5.8).** For instance, h-FH-Gittins with a constant horizon of $h = 50$, which provides nearly optimal perfor-

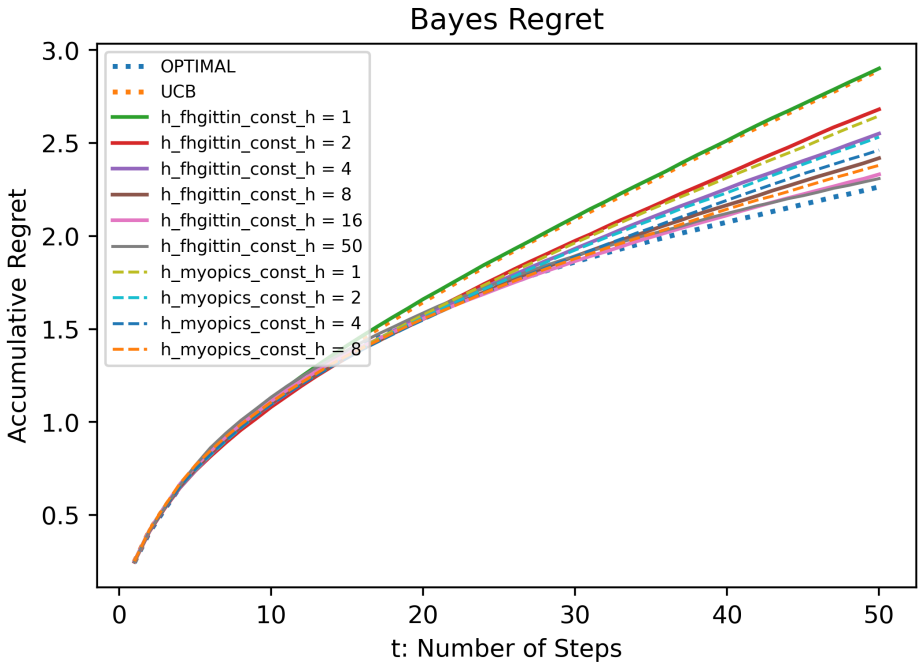

Figure 5.7: Comparison of Bayes regret between the constant horizon h-Myopic Optimal and constant horizon h-FH-Gittins strategies, with a true horizon of $T = 50$. For example, the parameter "h-myopic-const-h=4" represents the constant horizon h-myopic optimal policy with a horizon of 4.

mance, is faster than h-myopic optimal with a constant horizon of $h = 8$. **It is important to note that h-FH-Gittins is applicable only in the multi-armed bandit setting, whereas h-myopic optimal can be used in other learning settings, such as Reinforcement Learning.**

Note: Bayesian Q-learning method is included as one of our baselines but is not directly mentioned in any of the figures. As mentioned earlier, this method is equivalent to 2-myopic optimal with a constant horizon. Therefore, wherever you encounter "h-myopic-constant-h=2" in any figure, it refers to the Bayesian Q-learning baseline.

5.4 Conclusion

In conclusion, this chapter investigated two types of algorithms, namely h-FH-Gittins and h-myopic optimal, with two types of horizon approximation

methods: the constant horizon and the dynamic horizon. The results of the experiments provided valuable insights.

Firstly, it was observed that increasing the horizon leads to a closer performance to the Bayes-optimal policy, indicating that a longer planning horizon improves decision-making and reduces the Bayes' regret. Notably, even when overestimating the horizon (choosing $h > T$), the performance remained significantly better than underestimating it, highlighting the importance of considering a sufficiently large horizon.

Furthermore, the experiments revealed that the constant horizon h-FH-Gittins algorithm consistently outperforms the dynamic horizon h-FH-Gittins algorithm for all horizons. Similarly, the same trend was observed for the h-myopic optimal policy. This suggests that the constant horizon technique is a superior choice for horizon approximation in terms of achieving lower Bayes' regret.

Moreover, it was found that selecting a higher horizon is necessary for h-FH-Gittins to achieve the same level of performance as h-myopic optimal. However, this trade-off is justified by the fact that h-FH-Gittins is significantly faster than h-myopic optimal. It should be noted that h-FH-Gittins is specifically applicable to the multi-armed bandit setting, while h-myopic optimal can be employed in other learning settings, such as reinforcement learning.

Overall, the findings emphasize the significance of horizon selection in decision-making algorithms. By choosing an appropriate horizon and leveraging the advantages of the constant horizon technique, it is possible to achieve improved performance and faster computational efficiency in different learning settings. These results contribute to the understanding of horizon approximation methods and their impact on decision-making strategies.

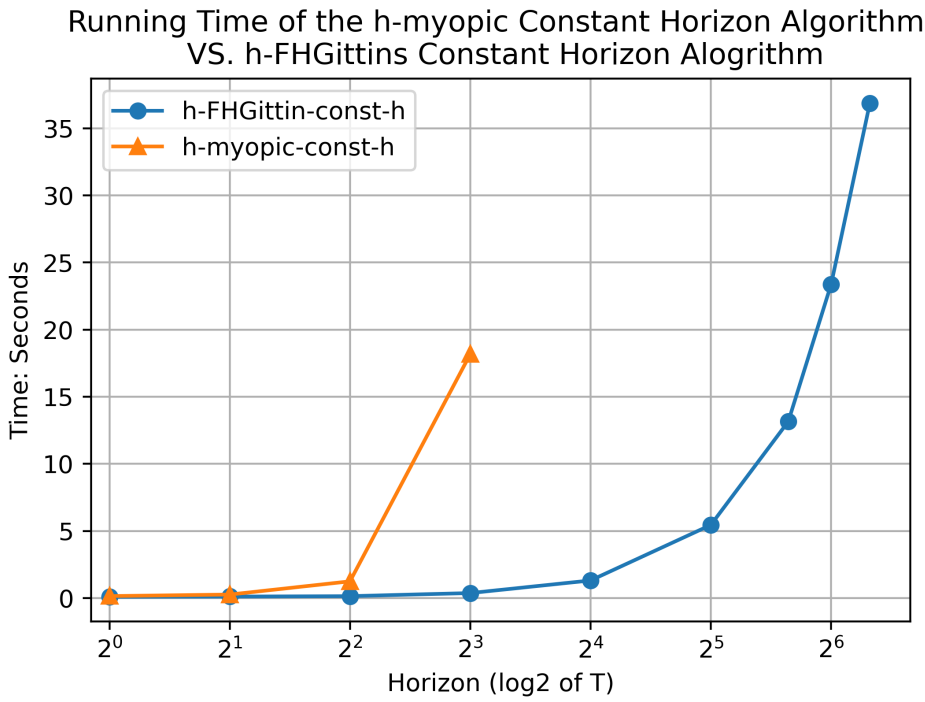

Figure 5.8: Time-wise comparison of "h-myopic-const-h" and "h-FHGittin-const-h" algorithms for different horizon approximation values.

Chapter 6

Conclusion

In our research to enhance the sample efficiency of Reinforcement Learning, we ventured through a broad landscape of domains and strategies. These strategies ranged from leveraging external advice to fine-tuning exploration strategies. The melding of Bayesian frameworks, multi-armed bandits, and advice-seeking provided a deeper understanding of decision-making processes. Here's a detailed recap of our main contributions and findings:

Exploration and Advice-Seeking: Central to our inquiry was how best to decide on the optimal arm for seeking advice and the right moment for such a query. The use of a Bayesian perspective led to a methodical approach to these questions, with the Value of Information (VOI) emerging as a crucial metric for evaluation.

Optimal Solution to "What arm to ask about." Our exploration in the "What Arm to Ask About" chapter ,Chapter 3, was instrumental in answering this question. Notably, the horizon approximation played a significant role in enhancing decision accuracy. However, it was the h-myopic approach that consistently demonstrated better performance over the Gittins' argmax method, mainly due to its superior horizon estimate, highlighting the importance of anticipating potential future rewards.

When to ask for advice. The study in the "When To Ask For Advice" chapter (Chapter 4) unveiled the optimal solution to this question. The research found that sometimes delaying advice-seeking can be more beneficial.

The distinction between the "optimal h" algorithm and the "approx h" method became apparent when considering their individual strategies, with the former showing clear advantages in our experiments.

Bayes-Optimal Exploration Strategy. The "Optimal Exploration" chapter (Chapter 5) investigated the h-FH-Gittins and h-Myopic Optimal algorithms in the context of horizon approximation. The results highlighted the role of horizon selection in decision-making algorithms, with a consistent preference for the constant horizon method over its dynamic counterpart.

Reflecting on the implications of our findings, they do extend beyond the theoretical realm. However, it's prudent to acknowledge that while they provide insights into RL's practical application, it's crucial to be judicious in interpreting their real-world applicability.

6.1 Future Work

The insights and results presented in this thesis naturally usher in several fascinating avenues for further research:

Adapting the Gittins' Index Policy. The "Optimal Exploration" chapter's results (Chapter 5) indicated a potential avenue for refining the Gittins' algorithm, especially for a finite horizon setting. Finding ways to minimize errors for the Gittins' argmax algorithm might further its practical use.

Enhanced Approximations. Given the computational challenges associated with certain problems, there's a pressing need for more sophisticated approximation strategies. This is especially relevant for the "What arm to ask about" and "When to ask for advice" dilemmas. Crafting approximations suitable for complex problem settings remains a worthwhile pursuit.

Complex Problem Settings and Advice Types. A promising direction would be evaluating the optimal "when to ask for advice" solution in more

elaborate problem scenarios. Investigating diverse and more intricate advice types can shed light on the tangible benefits of soliciting advice judiciously.

Broadening the Horizon. An extension of horizon approximation techniques to other learning scenarios, aside from multi-armed bandits, could be a valuable avenue in Reinforcement Learning.

While our research has covered substantial ground in RL sample efficiency, many more horizons await exploration, and we look forward to the continued journey in this research area.

References

[1] O. Amir, E. Kamar, A. Kolobov, and B. Grosz, "Interactive teaching strategies for agent training," in *In Proceedings of IJCAI 2016*, 2016.

[2] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[3] D. A. Berry and B. Fristedt, "Bandit problems: Sequential allocation of experiments (monographs on statistics and applied probability)," *London: Chapman and Hall*, vol. 5, no. 71-87, pp. 7–7, 1985.

[4] R. N. Bradt, S. Johnson, and S. Karlin, "On sequential designs for maximizing the sum of n observations," *The Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 1060–1074, 1956.

[5] F. L. Da Silva, P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "Uncertainty-aware action advising for deep reinforcement learning agents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 5792–5799.

[6] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," *Aaai/iaai*, vol. 1998, pp. 761–768, 1998.

[7] M. O. Duff, *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

[8] J. A. Edwards, *Exploration and exploitation in Bayes sequential decision problems*. Lancaster University (United Kingdom), 2016.

[9] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[10] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 41, no. 2, pp. 148–164, 1979.

[11] R. A. Howard, "Information value theory," *IEEE Transactions on systems science and cybernetics*, vol. 2, no. 1, pp. 22–26, 1966.

[12] E. T. Jaynes, *Probability theory: The logic of science*. Cambridge university press, 2003.

[13] E. Kaufmann, "Analyse de stratégies bayésiennes et fréquentistes pour l'allocation séquentielle de ressources," Ph.D. dissertation, Paris, ENST, 2014.

[14] D. E. Knuth, *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Addison-Wesley, 1968.

[15] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[16] D. B. Lawrence, *The economic value of information*. Springer Science & Business Media, 2012.

[17] M. Memarzadeh and M. Pozzi, "Value of information in sequential decision making: Component inspection, permanent monitoring and system-level scheduling," *Reliability Engineering & System Safety*, vol. 154, pp. 137–151, 2016.

[18] H. Raiffa, R. Schlaifer, *et al.*, "Applied statistical decision theory," 1961.

[19] R. Srinivasan and A. K. Parlikad, "Value of condition monitoring in infrastructure maintenance," *Computers & Industrial Engineering*, vol. 66, no. 2, pp. 233–241, 2013.

[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[21] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[22] L. Torrey and M. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 1053–1060.

Appendix A

Appendix

A.1 Proof of claim 1

Claim (1 Restated)**.**

$$\mathop{\mathbb{E}}_{\substack{x^a \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [q^b(n, b_{|x^a}, a^*(n, b))] = q^*(n, b, a^*(n, b)) = v^*(n, b)$$

Proof. Let j_t be the index for the supports of the probability distribution for the reward at time-step t, $h_t = \{x_{a_1^*} = r_{j_1}, x_{a_2^*} = r_{j_2}, ..., x_{a_t^*} = r_{j_t}\}$ be the history till time-step t subject to $h_0 = \{\}$, $a_t^* = a^*(n-(t-1), b_{|h_{t-1}})$, in which $a^*(n = T-(t-1), b)$ is the optimal action based on Bayes-optimal policy at time t.

Let's call the left hand-side of claim A.1 as (1) and the right hand-side of claim A.1 as (2). Now, let's expand the expression (1):

$$\mathop{\mathbb{E}}_{\substack{r \sim Pr(.|\theta_a) \\ \theta_a \sim b_a}} [q^b(n, b_{|x^a=r}, a_1^*)] = \sum_{i=1}^{|R|} b_a(r_i) q^b(n, b_{|x^a=r_i}, a_1^*) \tag{A.1}$$

In the above equation, $b_a(r) = P(r \mid b_a)$ is the probability of observing reward r for arm a based on the belief for arm a. Also, the expression $\sum_{i=1}^{|R|}$ iterate over all the possible rewards.

$$
\begin{aligned}
q^b(n, b_{|X_a=r_i}, a_1^*) &= \mathop{\mathbb{E}}_{\substack{r\sim \\ b_{a_1^*|x_a=r_i}}} [x_{a_1^*} + q^{b_{|x_{a_1^*}=r}}(n-1, b_{|x_a=r_i, x_{a_1^*}=r}, a_2^*)] \\
&\xrightarrow{expand\ 1\ step} = \sum_{j_1} b_{a_1^*|x_a=r_i}(r_{j_1})(r_{j_1} + q^{b_{|x_{a_1^*}=r_{j_1}}}(n-1, b_{|x_a=r_i, x_{a_1^*}=r_{j_1}}, a_2^*)) \\
&\xrightarrow{expand\ 2\ steps} = \sum_{j_1} b_{a_1^*|x_a=r_i}(r_{j_1})(r_{j_1} + \\
&\qquad \sum_{j_2} b_{a_2^*|x_a=r_i,h_1}(r_{j_2})(r_{j_2} + q^{b_{|h_2}}(n-2, b_{|x_a=r_i,h_2}, a_3^*))) \\
&\qquad \vdots \\
&\xrightarrow{expand\ n'\ steps} = \sum_{j_1} b_{a_1^*|x_a=r_i}(r_{j_1})(r_{j_1} + \ldots (r_{j_{n'-1}} + \\
&\qquad \sum_{j_{n'}} b_{a_{n'}^*|x_a=r_i,h_{n'-1}}(r_{n'})(r_{n'} + q^{b_{|h_{n'}}}(n', b_{|x_a=r_i,h_{n'}}, a_{n'+1}^*))))
\end{aligned}
\tag{A.2}
$$

Now Let's equation A.2 into equation A.1:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{r_i\sim b_a}[q^b(n, b_{|X_a}, a_1^*)] = \\
&\sum_i b_a(r_i) \sum_{j_1} b_{a_1^*|x_a=r_i}(r_{j_1})(r_{j_1} + \ldots (r_{j_{n'-1}} + \\
&\sum_{j_{n'}} b_{a_{n'}^*|x_a=r_i,h_{n'-1}}(r_{n'})(r_{n'} + q^{b_{|h_{n'}}}(n', b_{|x_a=r_i,h_{n'}}, a_{n'+1}^*))))
\end{aligned}
\tag{A.3}
$$

Now, let's expand the expression (2), with almost the same procedure as equation A.2:

$$
\begin{aligned}
q^*(n, b, a_1^*) &= q^b(n, b, a_1^*) \\
&\xrightarrow{expand\ utill\ t=n'} = \sum_{j_1} b_{a_1^*}(r_{j_1})(r_{j_1} + \ldots (r_{j_{n'-1}} + \\
&\qquad \sum_{j_{n'}} b_{a_{n'}^*|h_{n'-1}}(r_{n'})(r_{n'} + q^{b_{|h_{n'}}}(n', b_{|h_{n'}}, a_{n'+1}^*)))) \\
&\qquad = \sum_{j_1} b_{a_1^*}(r_{j_1})(r_{j_1} + \ldots (r_{j_{n'-1}} + \\
&\qquad \sum_{j_{n'}} b_{a_{n'}^*|h_{n'-1}}(r_{n'})(r_{n'} + q^*(n', b_{|h_{n'}}, a_{n'+1}^*))))
\end{aligned}
\tag{A.4}
$$

Let's define H_t as the random variable for the probability space over all possible histories with horizon t, and h_t is a sample from H_t. for any $t < n$,

$h_t \subset h_n$ if and only if h_t contains the first t observations of the h_n.

claim: All the roll-outs of expressions A.3 and A.4 are the same till a specific levels for each roll-out, and that level for each roll-out is just before action a is chosen.

proof: Due to the fact that the actions probability distributions are independent, we can see that if the action a is not chosen till, for example step n', all the b_{a*_i} for $i = 1, ..., n' - 1$ are independent of the information provided for action a. So, the roll-outs for both expressions A.3 and A.4 are the same till step n'. Note: it is worth mentioning that n' would be different for different roll-outs, depending on the actions taken previous steps $(a*_i,\ i < n')$

Now, Let's define a set, named $\mathcal{H}$, in which exists all the history samples (roll-outs), $h_t \subset h_n \in H_n$, such that in each of those samples, action a has not been taken up to observation step t-1 and the last observation is for action a. In other words, $\mathcal{H}$ contains all the possible roll-outs that are created in the same way described in the claim above.

$\forall h \in \mathcal{H}$, each with a unique size, measured with $n' = \text{size}(h)$, we can prove that the value for expression (1) and (2), in which the first n' steps are that of the observations of h, are equal.

Let consider one roll-out from $\mathcal{H}$ and call it $h' = r_{j_1}, r_{j_2}, \ldots, r_{j_{n'}}$. For this h' we can rewrite expression A.3 and A.4 as follows:

$$
\begin{aligned}
&expression A.3 = \\
&\sum_i b_a(r_i)(r_{j_1} + \ldots (r_{j_{n'-1}} + \sum_{j_{n'}} b_{a^*_{n'}|x_a=r_i,h_{n'-1}}(r_{n'})(r_{n'} + \\
&q^{b_{|h_{n'}}}(n', b_{|x_a=r_i,h_{n'}}, a^*_{n'+1})))) \\
&= (r_{j_1} + \ldots (r_{j_{n'-1}} + \sum_i b_a(r_i) \sum_{j_{n'}} b_{a|x_a=r_i,h_{n'-1}}(r_{n'})(r_{n'} + \\
&q^{b_{|h_{n'}}}(n', b_{|x_a=r_i,h_{n'}}, a^*_{n'+1})))) \\
&= (r_{j_1} + \ldots (r_{j_{n'-1}} + \sum_{j_{n'}} b_{a|h_{n'-1}}(r_{n'})(r_{n'} + q^{b_{|h_{n'}}}(n', b_{|x_a=r_i,h_{n'}}, a^*_{n'+1}))))
\end{aligned}
$$

$$expressionA.4 =$$
$$= (r_{j_1} + \ldots (r_{j_{n'-1}} + \sum_{j_{n'}} b_{a|h_{n'-1}}(r_{n'})(r_{n'} + q^{b_{|h_{n'}}}(n', b_{|h_{n'}}, a^*_{n'+1}))))$$

As you can see, these two rolled out expressions based on h' are equals. Now, given the fact that $\forall h \in \mathcal{H}$ the corresponding rolled out version of the expression (1) and (2), based on h', are the same, we can say that claim A.1 holds.

□

A.2 Proof of claim 5

Claim (5 Restated)**.** *In the context of a Bayesian decision-making process, where an agent seeks to maximize its expected cumulative reward over a decision horizon T, optimizing the Expected Value of Current Sample Information (E[VOCSI]) at each decision step maximizes the expected cumulative return. Specifically, if the agent, at each time step t, chooses to seek advice based on the maximization of E[VOCSI], then this strategy will lead to an optimal sequence of decisions that maximizes the overall expected return from the current time step to the end of the decision horizon.e*

Proof. To provide proof that maximizing the Expected Value of Current Sample Information (E[VOCSI]) at each time step maximizes the expected cumulative reward over the horizon, we need to represent the problem setting.

Problem Setting

- **Cumulative Reward**: Let x_t be the reward obtained at time t, and the cumulative reward over the horizon T is $\sum_{t=1}^{T} x_t$.
- **Expected Cumulative Reward**: The expected cumulative reward when following policy π starting at time t is $\mathop{\mathbb{E}}_{\pi}\left[\sum_{\tau=t}^{T} x_\tau \mid b_t\right]$, where b_t is the belief state at time t.
- **E[VOCSI]**: Let E[VOCSI$_t$] represent the expected value of current sample information if the agent seeks advice at time t.

Proof of Optimal Timing

1. **Decision at Each Time Step**: At each time t, the agent decides whether to seek advice. This decision is based on the comparison between $\mathbb{E}[VOCSI_t]$ and the expected gain from postponing advice.

2. **Impact of Advice**: Seeking advice at time t potentially alters the policy π from t onwards. Let's denote the policy after seeking advice at t as π'_t.

3. **Expected Gain from Advice**: The expected gain from seeking advice at time t can be written as:

$$\Delta_t = \mathbb{E}_{\pi'_t}\left[\sum_{\tau=t}^{T} R_\tau \mid b_t\right] - \mathbb{E}_{\pi}\left[\sum_{\tau=t}^{T} R_\tau \mid b_t\right] \tag{A.5}$$

4. **Maximizing E[VOCSI]**: We assert that Δ_t is maximized when $\mathbb{E}[VOCSI_t]$ is maximized, i.e., when the information obtained from seeking advice at time t leads to the greatest improvement in the expected cumulative reward from t to T.

5. **Mathematical Formulation**: To prove this, we need to show that:

$$t^* = \arg\max_t \mathbb{E}[VOCSI_t] \implies t^* = \arg\max_t \Delta_t \tag{A.6}$$

6. **Dynamic Consistency**: Given the fact that the formulation of Δ_T another variation of the z The decision to maximize $\mathbb{E}[VOCSI_t]$ at each time step is consistent with the principle of dynamic programming, asserting that an optimal policy consists of making the best decision at the current time step, given the optimal policies for future time steps.

Conclusion

If the relationship between $\mathbb{E}[VOCSI_t]$ and the expected gain in cumulative reward Δ_t is as stated, then by maximizing $\mathbb{E}[VOCSI_t]$ at each time step, the agent is effectively maximizing its expected cumulative reward over the horizon T.

□