Modern Dynamic Programming Approaches to Sequential Decision Making

Seungki Min

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Modern Dynamic Programming Approaches to Sequential Decision Making

Seungki Min

Dynamic programming (DP) has long been an essential framework for solving sequential decision-making problems. However, when the state space is intractably large or the objective contains a risk term, the conventional DP framework often fails to work. In this dissertation, we investigate such issues, particularly those arising in the context of multi-armed bandit problems and risk-sensitive optimal execution problems, and discuss the use of modern DP techniques to overcome these challenges such as information relaxation, policy gradient, and state augmentation. We develop frameworks formalize and improve existing heuristic algorithms (e.g., Thompson sampling, aggressive-in-the-money trading), while shedding new light on the adopted DP techniques.

Table of Contents

Acknowledgements

I would like to express my deep gratitude to the following people, without whom I would not have been able to complete my doctoral study. I thank my supervisors, Prof. Ciamac Moallemi and Prof. Costis Maglaras, for providing invaluable guidance in all aspects, the faculty members in the Decision, Risk, and Operations division, especially Prof. Daniel Russo, Prof. Yash Kanoria, Prof. Santiago Balseiro, and Prof. Mark Broadie, for sharing inspiring ideas, and my dear colleagues, Pengyu Qien, Muye Wang, and Alex Yu, for their friendship and encouragement. Finally, my special thanks go to my wife Jiyun Kim and our newborn daughter Chaewon Min.

Chapter 1: Introduction and Background

Dynamic programming (DP) has long been an essential framework for solving sequential decision-making problems in a wide variety of domains. In a DP framework, the stochastic environment that the decision maker (DM) encounters is described by a Markov decision process (MDP), and the optimal policy that maximizes the expected cumulative reward (or minimizes the expected cumulative cost) can be obtained by solving the Bellman equation analytically or numerically.

Despite its wide applicability, the DP approach often presents challenges in real-world applications. When the state space is continuous or intractably large, for example, the optimal policy cannot be implemented unless the Bellman equation admits an analytic solution. When the DM is not risk-neutral (e.g., the DM's objective is not just the expected reward but also contains some risk term), the Bellman's optimality principle is no longer valid. In what follows, we illustrate such challenges that arise in multi-armed bandit problems and risk-sensitive optimal execution problems, and discuss how to overcome these issues using modern DP techniques.

1.1 Bayesian multi-armed bandit problem and Thompson sampling algorithm

The multi-armed bandit (MAB) problem concerns a situation where the DM is given a set of arms with unknown reward distributions and decides which arm to select at each time so as to maximize the cumulative reward. This problem specifically highlights the issue that the DM has to find a balance between exploitation (i.e., selecting the currently known best arm) to maximize the immediate reward and exploration (i.e., selecting an arm that has not been tested enough) to maximize the informational gain. As the simplest instance of a reinforcement learning problem, the MAB problem has received enormous attention over the past decades.

In the earliest work in the bandit literature, the MAB problem was considered in a Bayesian setting and formulated as an MDP problem, in which the DM's belief on the unknown model parameters is interpreted as a state that evolves according to the Bayes' rule whenever the DM observes a reward realization. Under this MDP formulation, the optimal policy exists as a solution to the associated Bellman equation; e.g., the seminal work of [1] characterizes such an optimal policy in a discounted infinite horizon setting. However, the optimal policy is not feasible to implement in most cases, since the belief state space is intractably large (its size scales exponentially in the number of arms) and the Bellman equation cannot be solved explicitly.

We particularly focus on Thompson sampling (TS), which we understand as an approximate DP solution that effectively mitigates the curse of dimensionality discussed above. TS utilizes the idea of "posterior sampling"; i.e., at each decision epoch, it draws a random sample of the model parameters from the posterior distribution and selects the arm that is best given the sampled model parameters, i.e., it makes a decision as if the sampled model parameters are the ground truth. Due to its intuitive mechanism and computational efficiency, it has been enjoying tremendous success in practice and is being adopted and implemented by Google, Microsoft, Facebook, and many other firms in their daily operations.

However, TS often falls short of achieving a state-of-the-art performance as it does not explicitly take into account the value of exploration, i.e., its arm-selection rule does not consider how the DM's belief will change during the remaining time periods. This can be critical in practical settings, in which, for example, a time constraint restricts the amount of learning, or each action conveys a different amount of information, or the extra randomness in the system naturally leads to "free" exploration.

In this dissertation, we examine the use of two different DP techniques, namely, information relaxation and policy gradient, and develop two general frameworks that provide systematic ways to improve TS, with the aim of producing a better approximation of the Bayesian optimal policy.

- **Thompson sampling with information relaxation penalties** (Chapter 2). We first propose a framework that naturally generalizes TS by extending the idea of posterior sampling. An

algorithm in this framework draws a random sample of the entire future reward realizations in addition to the model parameters and decides which arm to pull by solving a deterministic reward maximization problem with respect to the sampled future scenario in the presence of penalties. We show that TS is a special case that follows from a particular penalty scheme and can be improved by incorporating penalties that reflect the value of future information more precisely.

- **Policy gradient optimization of Thompson sampling policies** (Chapter 3). We then propose a data-driven framework that can numerically optimize the control parameters of TS using the policy gradient method. While the policy gradient is a general tool for optimizing a randomized policy, it fails to work for TS since the likelihood of an arm being selected by TS cannot be written in a closed form in general. To overcome this issue, we interpret the sampled model parameters as a pseudo-action taken by TS, whose probability distribution is available in a closed form, and then apply the policy gradient in this pseudo-action space.

Comparison of above two frameworks shows that the one with information relaxation improves the performance of TS by investing additional online computation cost without need of extra control parameters, whereas the one with policy gradient does it by investing additional offline computation cost without need of application-specific analysis. Generally speaking, the former is more analytical and more suitable for situations where there exist some restrictions in the DM's decision making, whereas the latter is more practical and more widely applicable. Both frameworks leverage ideas developed in simulation literature and provide systematic ways to improve TS that achieve a more precise exploration–exploitation trade-off. We also provide theoretical analyses and numerical experiments showing that our suggested methodologies effectively fix the shortcomings of TS and achieve state-of-the-art performance in various settings.

1.2 Risk-sensitive optimal control problem via a conditional value-at-risk measure

In real-world applications, the DM often wants to be conservative in the face of uncertainties, by optimizing performance in adverse scenarios rather than focusing on average performance. This has long been an important topic in operations research and has been studied in the areas of risk-sensitive optimization and robust optimization.

In Chapter 4, we consider the use of conditional value-at-risk (CVaR) as an objective, which measures the average cost in a certain fraction of worst scenarios, i.e., the conditional average in the tail of the cost distribution. CVaR is particularly favored in practice also in theory because it offers a very intuitive quantification of uncertainty and also has nice mathematical properties. In the studies of optimal control under a risk measure, however, it has been considered difficult in general to apply the conventional DP framework due to the time-inconsistency of the risk measure (roughly speaking, a composition of CVaR measures is not a CVaR measure).

As an alternative, we leverage the idea of state augmentation that introduces an extra state variable representing the quantile value (at which the CVaR value of the future cost is measured), and develop a CVaR dynamic programming framework in the continuous-time setting. More specifically, we show that a certain type of CVaR-optimal control problem can be described as a continuous-time stochastic game between the DM who controls the original state process and an adversary who controls the quantile value process, based on the dual representation of the CVaR measure. We further derive a Bellman-like optimality equation that has a form of minimax optimization by exploiting the martingale representation theorem.

We adopt the suggested methodology to solve a "risk-sensitive optimal execution problem", given a task of liquidating a specific amount of a financial asset, the DM controls the liquidation rate adaptively to the price change so as to minimize the CVaR value of the total transaction cost, measured at a target quantile level. By solving partial differential equations that follow from the optimality equation, we derive the optimal dynamic trading strategy in a closed form, and characterize its "aggressiveness-in-the-money" behavior formally. An analytic comparison with

the optimized static trading strategy is also provided.

Chapter 2: Thompson Sampling with Information Relaxation Penalties

2.1 Introduction

Dating back to the earliest work [2, 1], multi-armed bandit (MAB) problems have been considered within a Bayesian framework, in which the unknown parameters are modeled as random variables drawn from a known prior distribution. In this setting, the problem can be viewed as a Markov decision process (MDP) with a state that is an information state describing the beliefs of unknown parameters that evolve stochastically upon each play of an arm according to Bayes' rule.

Under the objective of expected performance, where the expectation is taken with respect to the prior distribution over unknown parameters, the (Bayesian) optimal policy (OPT) is characterized by Bellman equations immediately following from the MDP formulation. In the discounted infinite-horizon setting, the celebrated Gittins index [1] determines an optimal policy, despite the fact that its computation is still challenging. In the non-discounted finite-horizon setting, which we consider, the problem becomes more difficult [3], and except for some special cases, the Bellman equations are neither analytically nor numerically tractable, due to the curse of dimensionality. In this paper, we focus on the Bayesian setting, and attempt to apply ideas from dynamic programming (DP) to develop tractable policies with good performance.

To this end, we apply the idea of *information relaxation* [4], a technique that provides a systematic way of obtaining the performance bounds on the optimal policy. In multi-period stochastic DP problems, admissible policies are required to make decisions based only on previously revealed information. The idea of information relaxation is to consider non-anticipativity as a constraint imposed on the policy space that can be relaxed, while simultaneously introducing a penalty for this relaxation into the objective, as in the usual Lagrangian relaxations of convex duality theory. Under such a relaxation, the decision maker (DM) is allowed to access future information and is asked

to solve an optimization problem so as to maximize her total reward, in the presence of penalties that punish any violation of the non-anticipativity constraint. When the penalties satisfy a condition (dual feasibility, formally defined in §2.3), the expected value of the maximal reward adjusted by the penalties provides an upper bound on the expected performance of the (non-anticipating) optimal policy.

The idea of relaxing the non-anticipativity constraint has been studied in different contexts [5, 6, 7, 8], and was later formulated as a formal framework by [4], upon which our methodology is developed. This framework has been applied to a variety of applications including optimal stopping problems [9]; linear-quadratic and linear-convex control [10, 11]; dynamic portfolio execution [12]; and more [e.g., 13, 14]. Typically, the application of this method to a specific class of MDPs requires custom analysis. In particular, it is not always easy to determine penalty functions that (1) yield a relaxation that is tractable to solve, and (2) provide tight upper bounds on the performance of the optimal policy. Moreover, the established information relaxation theory focuses on upper bounds and provides no guidance on the development of tractable policies.

Our contribution is to apply the information relaxation techniques to the finite-horizon stochastic MAB problem, explicitly exploiting the structure of a Bayesian learning process. In particular,

1. we propose a series of information relaxations and penalties of increasing computational complexity;
2. we systematically obtain the upper bounds on the best achievable expected performance that trade off between tightness and computational complexity;
3. and we develop associated (randomized) policies that generalize Thompson sampling (TS) in the finite-horizon setting.

In our framework, which we call *information relaxation sampling*, each of the penalty functions (and information relaxations) determines one policy and one performance bound given a particular problem instance specified by the time horizon and the prior beliefs. As a base case for our algorithms, we have TS [15] and the conventional regret benchmark that has been used for Bayesian regret analysis since [16]. At the other extreme, the optimal policy OPT and its expected

performance follow from the "ideal" penalty (which, not surprisingly, is intractable to compute). By picking increasingly strict information penalties, we can improve the policy and the associated bound between the two extremes of TS and OPT.

As an example, one of our algorithms, IRS.FH, is a very simple modification of TS that naturally incorporates time horizon T. Recalling that TS makes a decision based on sampled parameters for each arm from the posterior distribution in each epoch, observe that knowledge of the parameters is essentially (assuming Bayesian consistency) as informative as having an infinite number of future reward observations from each arm. By contrast, IRS.FH makes a decision based on future Bayesian estimates, updated with only $T - 1$ future reward realizations for each arm, where the rewards are sampled based on the inital posterior belief. When $T = 1$ (equivalently, at the last decision epoch), such a policy takes a myopically best action based only on the current estimates, which is indeed an optimal decision, whereas TS would still explore unnecessarily. While keeping the recursive structure of the sequential decision-making process of TS, IRS.FH naturally performs less exploration than TS as the remaining time horizon diminishes. This mitigates a common practical criticism of TS: it explores too much.

Beyond this, we propose other algorithms that more explicitly quantify the benefit of exploration and more explicitly trade off between exploration and exploitation, at the cost of additional computational complexity. As we increase the complexity, we achieve policies that improve performance, and separately provide tighter tractable computational upper bounds on the expected performance of any policy for a particular problem instance. By providing natural generalizations of TS, our work provides both a deeper understanding of TS and improved policies that do not require tuning. Since TS has been shown to be asymptotically regret optimal in some settings, e.g., by the metric of growth-rate [17] or by the metric of worst-case regret [18, 19], our improvements can at best be (asymptotically) constant factor improvements by that metric. On the other hand, TS is extremely popular in practice, and we demonstrate in numerical examples that the improvements can be significant and are likely to be of practical interest.

Moreover, we develop upper bounds on performance that are useful in their own right. Suppose

that a decision maker faces a particular problem instance and is considering any particular MAB policy (be it one we suggest or otherwise). By simulating the policy, we can find a lower bound on the performance of the optimal policy. We introduce a series of upper bounds that can also be evaluated in any problem instance via simulation. Paired with the lower bound, these provide a computational, simulation-based "confidence interval" that can be helpful to the decision maker. For example, if the upper bound and lower bound are close, the suboptimality gap of the policy under consideration is guaranteed to be small, and it is not worth investing in better policies.

2.2 Problem

2.2.1 Bayesian MAB with independent arms

We consider a Bayesian MAB problem with K *independent arms* and a *finite time horizon* T. More specifically, we define an MAB instance with a tuple $\left(K, T, \mathcal{R}, \Theta, \mathcal{P}, \mathcal{Y}, \mathbf{y}\right)$ as follows. In each period $t = 1, \ldots, T$, the decision maker (DM) selects one among K arms, each of which yields a stochastic reward whenever selected. We let $\mathcal{A} \triangleq \{1, \ldots, K\}$ denote the set of arms, and let $R_{a,n}$ denote the random variable that represents the reward from the n^{th} pull[1] of arm $a \in \mathcal{A}$. For each arm a, the rewards $\{R_{a,n}\}_{n \in \mathbb{N}}$ are independent and identically distributed according to the distribution $\mathcal{R}_a(\theta_a)$, where $\theta_a \in \Theta_a$ is the *parameter* associated with arm a:

$$R_{a,n} \sim \mathcal{R}_a(\theta_a), \quad \forall n \in \mathbb{N}, \quad \forall a \in \mathcal{A}. \tag{2.1}$$

The parameter θ_a is unknown to the DM, and is modeled as a random variable for which we have a family of *conjugate priors* $\{\mathcal{P}_a(y_a)\}_{y_a \in \mathcal{Y}_a}$, i.e., a space of distributions for θ_a that is closed under a Bayesian update with a reward realization $R_{a,n}$. Given a *hyperparameter* $y_a \in \mathcal{Y}_a$ (also called a *belief*), consider a probability measure $\mathbb{P}_{y_a}[\cdot]$ under which the parameter θ_a follows the *prior*

[1] One may consider an alternative stochastic model for the reward realization process in which the rewards are defined through a time index (e.g., $R_{a,t}$ denotes the reward from arm a in period t). This would be mathematically equivalent from the perspective of the DM. However, once the information set is relaxed, such a model is *not* equivalent to ours: in our model, the DM is not allowed to skip any future reward realizations, and this is crucial for some of the algorithms suggested in this paper. See the discussion in §2.3.3.

distribution $\mathcal{P}_a(y_a)$:

$$\theta_a \sim \mathcal{P}_a(y_a), \quad \forall a \in \mathcal{A}. \tag{2.2}$$

Let $\mathbb{E}_{y_a}[\cdot]$ denote the expected value under this probability measure. For brevity, denote the vector of parameters and hyperparameters across arms by $\boldsymbol{\theta} \triangleq (\theta_1, \ldots, \theta_K)$ and $\mathbf{y} \triangleq (y_1, \ldots, y_K)$, respectively. Define $\mathcal{R}$, Θ, $\mathcal{P}$, $\mathcal{Y}$, $\mathbb{P}_{\mathbf{y}}$, and $\mathbb{E}_{\mathbf{y}}$ analogously. We will often describe an MAB instance only with a tuple $(T, \mathbf{y})$ when the other components are clear in context.

Throughout the paper, we assume that the rewards are absolutely integrable for each hyperparameter $y_a \in \mathcal{Y}_a$:

$$\mathbb{E}_{y_a}\left[|R_{a,1}|\right] < \infty, \quad \forall y_a \in \mathcal{Y}_a,\ a \in \mathcal{A}, \tag{2.3}$$

where the expectation is taken with respect to the random realization of the parameter θ_a and also with respect to the random realization of the reward $R_{a,1}$.

We further define the *outcome* $\omega \in \Omega$ (also referred to as the future or scenario) as a combination of the parameters and all future reward realizations, i.e.,

$$\omega \triangleq \left(\boldsymbol{\theta}, (R_{a,n})_{a\in\mathcal{A}, n\in\mathbb{N}}\right) \ \sim \ \mathcal{I}(\mathbf{y}), \tag{2.4}$$

that encodes all the uncertainties that the DM encounters in the environment and whose distribution is denoted by $\mathcal{I}(\mathbf{y})$.

Policy. Given an outcome ω, the reward at time t can be represented as a function of the DM's action sequence $\mathbf{a}_{1:t} = (a_1, \ldots, a_t) \in \mathcal{A}^t$, i.e.,

$$r_t(\mathbf{a}_{1:t}, \omega) \triangleq R_{a_t, n_t(\mathbf{a}_{1:t}, a_t)}, \tag{2.5}$$

where $n_t(\mathbf{a}_{1:t}, a) \triangleq \sum_{s=1}^{t} \mathbf{1}\{a_s = a\}$ counts how many times an arm a has been played up to time t (inclusive). Consequently, we define the *history* $H_t(\mathbf{a}_{1:t}, \omega)$ as the information revealed to the

DM up to time t when taking an action sequence $\mathbf{a}_{1:t}$ given the outcome ω:

$$H_t(\mathbf{a}_{1:t}, \omega) \triangleq \big(a_1, r_1(a_1, \omega), a_2, r_2(\mathbf{a}_{1:2}, \omega), \ldots, a_t, r_t(\mathbf{a}_{1:t}, \omega)\big). \tag{2.6}$$

Let $\mathbf{A}^{\pi}_{1:t}$ be the action sequence taken under the DM's policy π. We can define the natural filtration $\mathbb{F} \triangleq (\mathcal{F}_t)_{t=0,1,\ldots,T}$ where $\mathcal{F}_t \triangleq \sigma\left(H_t(\mathbf{A}^{\pi}_{1:t}, \omega)\right)$ is the σ-field generated by the history H_t.

A policy π is called *non-anticipating* if every action A^{π}_t is measurable with respect to $\mathcal{F}_{t-1}$; i.e., each decision is made based only on the information revealed prior to that time. We denote by $\Pi_{\mathbb{F}}$ the set of all non-anticipating policies, including randomized ones. The (Bayesian) *performance* of a policy π is measured by the total reward that π earns on average, i.e.,

$$V(\pi, T, \mathbf{y}) \triangleq \mathbb{E}_{\mathbf{y}}\left[\sum_{t=1}^{T} r_t(\mathbf{A}^{\pi}_{1:t}, \omega)\right], \tag{2.7}$$

where T and $\mathbf{y}$ specify, respectively, the length of the time horizon and the prior hyperparameters of given the MAB instance.

Bayesian update. Whenever the DM observes a reward realization, as a Bayesian learner, she can update her belief associated with the selected arm according to Bayes' rule. More formally, we introduce a *Bayesian update function* $\mathcal{U}_a : \mathcal{Y}_a \times \mathbb{R} \to \mathcal{Y}_a$ so that after observing a reward $r \in \mathbb{R}$ from an arm $a \in \mathcal{A}$, the hyperparameter associated with arm a is updated from y_a to $\mathcal{U}_a(y_a, r)$ (e.g., if $\theta_a \sim \mathcal{P}_a(y_a)$, then $\theta_a | R_{a,1} \sim \mathcal{P}_a(\mathcal{U}_a(y_a, R_{a,1}))$). We will often use $\mathcal{U} : \mathcal{Y} \times \mathcal{A} \times \mathbb{R} \to \mathcal{Y}$ to denote the updating of the hyperparameter vector $\mathbf{y}$; i.e., after observing a reward realization r from an arm a, the hyperparameter vector is updated from $\mathbf{y}$ to $\mathcal{U}(\mathbf{y}, a, r)$, where only the a^{th} component is updated.

We further describe the time evolution of the DM's belief throughout the decision making process. Given an outcome ω and an action sequence $\mathbf{a}_{1:t}$, the posterior hyperparameter vector at time t can be recursively expressed as

$$\mathbf{y}_t(\mathbf{a}_{1:t}, \omega; \mathbf{y}) \triangleq \mathcal{U}\left(\mathbf{y}_{t-1}(\mathbf{a}_{1:t-1}, \omega; \mathbf{y}), a_t, r_t(\mathbf{a}_{1:t}, \omega)\right), \quad \forall t \geq 1, \tag{2.8}$$

with $\mathbf{y}_0 \triangleq \mathbf{y}$. We often write $[\mathbf{y}_t(\mathbf{a}_{1:t}, \omega; \mathbf{y})]_a$ to denote the a^{th} component of $\mathbf{y}_t(\mathbf{a}_{1:t}, \omega; \mathbf{y})$. This hyperparameter vector $\mathbf{y}_t(\mathbf{a}_{1:t}, \omega; \mathbf{y})$ sufficiently describes the DM's belief given the history $H_t(\mathbf{a}_{1:t}, \omega)$.

Mean reward. We introduce several notions of mean reward that play a crucial role throughout the paper. For each arm $a \in \mathcal{A}$, we let $\mu_a(\theta_a)$ denote the *conditional mean reward* given the parameter θ_a, and let $\bar{\mu}_a(y_a)$ be the *predictive mean reward* given the hyperparameter y_a:

$$\mu_a(\theta_a) \triangleq \mathbb{E}\left[R_{a,n} \middle| \theta_a\right], \quad \bar{\mu}_a(y_a) \triangleq \mathbb{E}_{y_a}\left[\mu_a(\theta_a)\right]. \tag{2.9}$$

We further define the *posterior predictive mean reward process* $\left\{\hat{\mu}_{a,n}\right\}_{n\geq 0}$ by

$$\hat{\mu}_{a,n}(\omega; y_a) \triangleq \mathbb{E}_{y_a}\left[\mu_a(\theta_a) | R_{a,1}, \ldots, R_{a,n}\right], \tag{2.10}$$

which represents the predictive mean reward (i.e., the finite-sample Bayesian estimate of $\mu_a(\theta_a)$) after observing first n rewards associated with the arm a.

Remark 2.2.1. *Fix an arm $a \in \mathcal{A}$. The posterior predictive mean reward process $\{\hat{\mu}_{a,n}\}_{n\geq 0}$ is a martingale adapted to the filtration generated by the sequence of rewards $(R_{a,1}, R_{a,2}, R_{a,3}, \ldots)$. Furthermore, it starts at the value of the prior predictive mean reward $\bar{\mu}_a(y_a)$ and converges to the conditional mean reward $\mu_a(\theta_a)$; i.e., $\hat{\mu}_{a,0}(\omega; y_a) = \bar{\mu}_a(y_a)$ and $\lim_{n\to\infty} \hat{\mu}_{a,n}(\omega; y_a) = \mu_a(\theta_a)$ almost surely (see Proposition A.4.2 in the Appendix).*

2.2.2 Natural exponential family

We will often consider the case where the reward distribution $\mathcal{R}_a(\theta_a)$ belongs to the *natural exponential family*. In this case, the closed-form expressions are available for the aforementioned notation. For any given $\theta_a \in \Theta_a \subseteq \mathbb{R}$, the probability measure for a random reward $R_{a,n}$ is determined by

$$\mathbb{P}\left[R_{a,n} \in dr \mid \theta_a\right] = h_a(dr) \exp\left(\theta_a r - A_a(\theta_a)\right), \tag{2.11}$$

where $h_a(dr)$ is the *reference measure* and $A_a(\cdot)$ is the *log-partition function* that is a logarithm of the normalization factor. We then have a family of conjugate priors $\{\mathcal{P}_a(y_a)\}_{y_a \in \mathcal{Y}_a}$ where $\mathcal{Y}_a \triangleq \{y_a = (\xi_a, \nu_a) | \xi_a \in \mathbb{R}, \nu > 0\}$, so, for any given hyperparameter $y_a \in \mathcal{Y}_a$, the corresponding prior $\mathcal{P}_a(y_a)$ is also an exponential family distribution and can be described as

$$\mathbb{P}_{(\xi_a, \nu_a)} [\theta_a \in d\theta] = f_a(\xi_a, \nu_a) \exp(\xi_a \theta - \nu_a A_a(\theta)) \, d\theta, \tag{2.12}$$

where $f_a(\xi_a, \nu_a)$ is the normalization factor and ν_a represents the effective number of observations. Within this family of conjugate priors, it is well known that the posterior distribution can be expressed as

$$\mathbb{P}_{(\xi_a, \nu_a)} \left[\theta_a \in d\theta \,\middle|\, R_{a,1}, \ldots, R_{a,n}\right] = \mathbb{P}_{\left(\xi_a + \sum_{i=1}^n R_{a,i},\ \nu_a + n\right)} [\theta_a \in d\theta] \,. \tag{2.13}$$

This property can also be expressed via the Bayesian update function as $\mathcal{U}_a((\xi_a, \nu_a), r) = (\xi_a + r, \nu_a + 1)$. We also have the following identities for the mean reward metrics:

$$\mu_a(\theta_a) = A_a'(\theta_a), \quad \bar{\mu}_a(\xi_a, \nu_a) = \frac{\xi_a}{\nu_a}, \quad \hat{\mu}_{a,n}(\omega; \xi_a, \nu_a) = \frac{\xi_a + \sum_{i=1}^n R_{a,n}}{\nu_a + n}, \tag{2.14}$$

where $A_a' \triangleq dA_a / d\theta_a$. We refer the reader to [20] for further details.

Bernoulli and Gaussian MABs. We briefly illustrate the Bernoulli MAB and Gaussian MAB as representative examples of the problem instance described by a natural exponential family. In the Bernoulli MAB, the rewards of an arm are Bernoulli random variables whose success probability is drawn from a Beta distribution. In the Gaussian MAB, the rewards of an arm are normally distributed with an unknown mean and a known noise variance where the mean is also normally distributed. Table 2.1 summarizes the previously defined notation.

	Bernoulli MAB	Gaussian MAB
Prior distribution	$\mu_a \sim \text{Beta}(\alpha_a, \beta_a)$	$\mu_a \sim \mathcal{N}(m_a, v_a^2)$
Reward distribution	$R_{a,n} \sim \text{Bernoulli}(\mu_a)$	$R_{a,n} \sim \mathcal{N}(\mu_a, \sigma_a^2)$
Parameter θ_a	$\theta_a = \log \frac{\mu_a}{1-\mu_a}$	$\theta_a = \frac{\mu_a}{\sigma_a^2}$
Hyperparameters ξ_a, ν_a	$\xi_a = \alpha_a$, $\nu_a = \alpha_a + \beta_a$	$\xi_a = \frac{m_a \sigma_a^2}{v_a^2}$, $\nu_a = \frac{\sigma_a^2}{v_a^2}$
Reference measure h_a	$h_a(dr) = \delta_0(dr) + \delta_1(dr)$	$h_a(dr) = \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp\left(-\frac{r^2}{\sigma_a^2}\right) dr$
Log-partition function A_a	$A_a(\theta_a) = \log\left(1 + e^{\theta_a}\right)$	$A_a(\theta_a) = \frac{\sigma_a^2 \theta_a^2}{2}$
Mean reward μ_a	$\mu_a(\theta_a) = \frac{e^{\theta_a}}{1+e^{\theta_a}}$	$\mu_a(\theta_a) = \sigma_a^2 \theta_a$
Predictive mean $\bar{\mu}_a$	$\bar{\mu}_a(\alpha_a, \beta_a) = \frac{\alpha_a}{\alpha_a + \beta_a}$	$\bar{\mu}_a(m_a, v_a^2) = m_a$

Table 2.1: Description of a Bernoulli MAB and a Gaussian MAB. Here, $\delta_x(dr)$ denotes a Dirac measure that has a single atom at x.

2.2.3 Bayesian optimal policy

In a Bayesian framework, the MAB problem can be viewed as a Markov decision process (MDP) in which a state corresponds to an information state (or belief state) of the DM. It has the following recursive structure that we will exploit throughout the paper. Given an MAB instance with time horizon T and prior belief $\mathbf{y}$, suppose that the DM has just earned r by pulling an arm a at time $t = 1$. Then the remaining problem for the DM is equivalent to an MAB instance with time horizon $T - 1$ and prior belief $\mathcal{U}(\mathbf{y}, a, r)$. Based on this Markovian structure, we obtain the following Bellman equations for the MAB problem: for all $T \in \mathbb{N}$ and $\mathbf{y} \in \mathcal{Y}$,

$$Q^*(T, \mathbf{y}, a) \triangleq \mathbb{E}_{\mathbf{y}}\left[R_{a,1} + V^*(T-1, \mathcal{U}(\mathbf{y}, a, R_{a,1}))\right], \tag{2.15}$$

$$V^*(T, \mathbf{y}) \triangleq \max_{a \in \mathcal{A}} Q^*(T-1, \mathbf{y}, a), \tag{2.16}$$

with $V^*(0, \mathbf{y}) \triangleq 0$ for all $\mathbf{y} \in \mathcal{Y}$. The value function $V^*(T, \mathbf{y})$ represents the best possible performance that a non-anticipating policy can achieve in the MAB problem specified by the time

horizon T and the prior belief $\mathbf{y}$, or equivalently, the maximum expected future reward that one can earn during T remaining periods[2] when the current belief is $\mathbf{y}$.

While Bellman equations are, in general, intractable to solve and directly apply, they offer a characterization of the *Bayesian optimal policy* (OPT). At a certain moment, when the remaining time horizon is T and the belief is $\mathbf{y}$, OPT takes an action with the largest state-action value (Q-value), i.e., pulls the arm $A^* = \operatorname{argmax}_a Q^*(T, \mathbf{y}, a)$, and this action selection procedure is repeated while updating T and $\mathbf{y}$ according to Bayes' rule as described in Algorithm 1. Such a policy achieves the best possible performance among all non-anticipating policies:

$$V^*(T, \mathbf{y}) = \sup_{\pi \in \Pi_{\mathbb{F}}} V(\pi, T, \mathbf{y}) = V(\text{OPT}, T, \mathbf{y}), \quad \forall T \in \mathbb{N}, \mathbf{y} \in \mathcal{Y}. \tag{2.17}$$

Algorithm 1: Bayesian optimal policy (OPT)

Function `OPT` $(T, \mathbf{y})$

```
// T:remaining time horizon, y:current belief
```

1 **return** $\operatorname{argmax}_a Q^*(T, \mathbf{y}, a)$

Procedure `OPT-Outer` $(T, \mathbf{y})$

```
// T:time horizon, y:prior belief
```

1 $\mathbf{y}_0 \leftarrow \mathbf{y}$

2 **for** $t = 1, 2, \ldots, T$ **do**

3 Select $A_t \leftarrow$ `OPT` $(T - t + 1, \mathbf{y}_{t-1})$

4 Earn and observe a reward r_t and update belief $\mathbf{y}_t \leftarrow \mathcal{U}(\mathbf{y}_{t-1}, A_t, r_t)$

end

[2]We intentionally refrain from indexing the value function V^* by time t, since such a representation conceals the Markovian structure of the Bayesian MAB problem and leads to complicated expressions for the variables that exploit this Markovian structure. To avoid confusion, the horizon T will be written as an argument to functions whereas the time index t will be written as a subscript, throughout the paper.