論文 / 著書情報
Article / Book Information

題目(和文)	Online decision making in non-stationary Markovian environments
Title(English)	Online decision making in non-stationary Markovian environments
著者(和文)	MaYao
Author(English)	Yao Ma
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第10040号, 授与年月日:2015年12月31日, 学位の種別:課程博士, 審査員:杉山 将,徳永 健伸,篠田 浩一,村田 剛志,藤井 敦
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10040号, Conferred date:2015/12/31, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Online Decision Making in Non-stationary Markovian Environments

Yao Ma

December 2015

Department of Computer Science
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Thesis Committee:
Masashi Sugiyama, Chair
Takenobu Tokunaga
Koichi Shinoda
Tsuyoshi Murata
Atsushi Fujii

Submitted in partial fulfillment of
the requirements for the degree of

Doctor of Engineering

Keywords: Markov decision process, reinforcement learning, online learning, regret minimization

To my family

Abstract

Learning in non-stationary environments gains extensive attention since environments are not always fixed in reality. The tasks are more challenging due to the lack of information about the future evolution of environments. A plenty of problems concern how to make decisions with these challenging non-stationary environments. This thesis contributes to the theory of online decision making problems in non-stationary Markovian environments. Two different algorithms are proposed, which are aimed at solving online Markov decision process (MDP) problems with a large or continuous state space.

Firstly, we settle online MDP problems with continuous state and action spaces and propose the online policy gradient (OPG) algorithm. Although previously proposed online MDP algorithms have achieved some exciting results, these algorithms are not extensible to the continuous setting without additional assumptions. The proposed OPG algorithm solves the continuous problems with a parameterized policy model. This is the first work to give an online MDP algorithm that can handle continuous state and action spaces with guarantee. Through theoretical analyses, we show the proposed OPG algorithm is a no-regret algorithm in different scenarios. Furthermore, we demonstrate the experimental behavior of the OPG algorithm, which substantiates the theoretical results.

Secondly, we investigate the large (possibly infinite) state space problems and propose online Markov decision processes with policy iteration (OMDP-PI) algorithm. Compared with the state-of-the-art algorithms, the proposed algorithm aims at large state space online MDP problems where less computational complexity and the function approximation are necessary. To this end, the proposed OMDP-PI algorithm is motivated by the idea of combining the function approximation with policy iteration. We prove that our proposed OMDP-PI algorithm achieves a sub-linear regret with respect to a policy set. A significant benefit of the OMDP-PI algorithm is that a linear approximation could be used together with the OMDP-PI algorithm for large (continuous) state space problems, where the convergence is guaranteed. Through a grid world experiment, we illustrate the experimental performance of the OMDP-PI algorithm, which verifies the theoretical regret analysis.

Given the solid theoretical results, we conclude that the proposed algorithms could handle online MDP problems with large (continuous) state spaces.

Acknowledgments

First of all, I would like to acknowledge my supervisor Professor Masashi Sugiyama, who has inspired and guided me to complete my PhD study and my research. I extend sincere and deep gratitude for his support, guidance, encouragement, and everything over the past three years. I have learned a lot from him including research skills, writing skills, and immense knowledge. The most important thing I learned from him is the passion for doing good research.

I would like to thank all the other members of my thesis committee: Professor Takenobu Tokunaga, Professor Koichi Shinoda, Professor Tsuyoshi Murata, and Professor Atsushi Fujii, for their great suggestions and insightful comments.

Working at Sugiyama Lab was one of the most wonderful time in my life. I thank all my colleagues and ex-colleagues from Tokyo Institute of Technology and the University of Tokyo for all the fun we have had in the last three years.

The work in this thesis was financially supported by the Japanese government MEXT scholarship. I thank the Ministry of Education, Culture, Sports, Science and Technology for its generous financial support.

Finally, I would like to thank my family for letting me pursue my dream for so long so far away from home. I am extremely grateful to their endless love and support through the past three years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online decision making is a sequential interaction between a decision maker and a time varying environment. How can we learn the best strategy which gains the maximum benefit or minimum cost in this unknown changing environment without any priority knowledge? This thesis considers such problems with some specific environments, which are mathematically formalized as the *Markov decision processes* (MDPs). The difficulty of online decision making problems is the uncertainty of the environments, which means that the decision maker lacks of the future information about these changing environments. In this chapter, we introduce the motivation and definition of the problems involved in this thesis. A general description of online decision making is provided in Section 1.1. Then two mathematical frameworks for modeling decision making and the objective are presented in Section 1.2, Section 1.3, and Section 1.4, respectively. In Section 1.5 and Section 1.6, we introduce the contributions and the outline of this thesis.

1.1 Online Decision Making

Decision making routinely involves choice among temporally extended courses of action over a broad range of time scales (Sutton et al., 1999). Human make decisions for solving problems with solutions deemed to be satisfactory. Several perspectives have been mainly studied with respect to the decision making problems, such as psychological, cognitive, and normative. Besides knowing the mechanism of human decision making, an important perspective is to investigate how to imitate the decision making process by computers. In order to make the decision automatically, the computer needs to train a model which can make a good decision by observations. Most machine learning algorithms (e.g., supervised learning, unsupervised learning) aim at training models by collecting training data from environments. However, a significant advantage of human decision making is the capability of adaptation to the environmental changes including unseen situations (Kelemen et al., 2002). Therefore, the computer may learn to make better (human-like) decisions, which is adaptable to the changing environment. For this purpose, an online version of decision making is concerned to handle the decision making problem with non-stationary environments. *Online decision making* is a sequential decision making problem without the knowledge of the evolution of the future (Kalai and Vempala, 2005; Kleinberg, 2005). Over some time period, an online decision making problem involves an agent which could be either a human or a computer. The agent chooses one element from a set of alternatives in a changing environment. By the changing environment, we mean the environment decides the gain and consequence of the chosen decision. The main difficulty of the online decision making problem is to face the uncertainty, which has been well-studied in several topics by using the *exploration and exploitation* techniques. Kleinberg (2005) defined the online decision making problem with a quadruple $\{\mathcal{X}, \mathcal{C}, \Xi, \mathcal{F}\}$, where

- $(\mathcal{X}, \mathcal{C})$ is the online decision domain, where $\mathcal{X}$ is the set of strategies and $\mathcal{C} : \mathcal{X} \to \mathbb{R}$ is a class of cost functions.
- The feedback model is specified by a set Ξ and a function $\mathcal{F} : \mathcal{X} \times \mathcal{C} \to \Xi$. Ξ is the set of the feedback values, and $\mathcal{F}(x, c)$ is the feedback the player achieved when playing strategy x against cost function c.

In machine learning, most online decision problems concern the decision domain $(\mathcal{X}, [0,1]^{\mathcal{X}})$, where $\mathcal{C} = [0,1]^{\mathcal{X}}$ is the set of all mappings from $\mathcal{X}$ to $[0,1]$. A typical problem is called the *multi-armed bandit* problem, which is motivated by a gambler. Suppose there is a row of slot machines, where each of them is assigned with a random reward distribution. The player needs to choose one of the machines at each time step, which maximizes the total outcomes. Another well-studied problem is called *best-expert* problem, whose goal is to predict the best decision by choosing the best expert's prediction within a set of experts.

By the type of the non-stationary environments, we could consider two types of uncertainties: the uncertainty of the environment dynamics and the uncertainty of the outcomes. In this thesis, we mainly focus on facing the uncertainty of the outcomes. And the environment dynamics are assumed to be fixed and Markovian. According to the uncertainty of the outcomes, the online decision making problem can be divided into three types: stochastic, adversarial, and pre-fixed. In the stochastic online decision making problem, the outcomes are assumed to be determined by a latent variable which is drawn independently from an identical distribution. In the adversary scenario, we do not make any assumption on the evolution of the changing environment. The pre-fixed online decision making problem is in the middle of these two problems, where the outcomes are assume to be affected only by the current status and the changes could be either stochastic or adversarial. In this thesis, we mainly focus on the last scenario which usually can be formulated as Markov decision processes (MDPs).

1.2 Markov Decision Processes

The research of MDPs was known at least as early as the 1950s (Bellman, 1957). A central relevance resulted from the introduction of the dynamic programming concept by Bellman (1957). MDPs offer an exquisite mathematical model for dealing with sequential decision problems in which outcomes are partially controlled by a decision maker. MDPs are shown to be useful for solving a wide range of optimization problems that arise in the fields of operations research, automatic control, artificial intelligence, management science, finance, computer science, and others (Yu, 2006). Thanks to MDPs, researchers are capable of analyzing the dynamics of a stochastic process whose transition mechanism is controlled over time. Especially, automatic control and artificial intelligence are the most important fields where MDPs are widely used for solving problems. In automatic control, MDPs are used to solve nonlinear stochastic optimal control problems and adaptive optimal control problems (Busoniu et al., 2010). In artificial intelligence, MDPs are used to help the artificial agent learn how to behave in an unknown environment without requiring prior knowledge (Sutton and Barto, 1998).

The goal of the MDP algorithms is to choose decisions which perform best over an extended period of time. To reach the goal, there are two types of learning methods (Busoniu et al., 2010):

- *Dynamic programming (DP)*: It is an algorithmic method for solving MDP problems, which requires a model of the environment. The DP algorithms work offline, learning the best strategy of making decisions which is then used to control the process. Usually, there is no need to obtain the analytic solution for the problem. DP is used for such a generative model which is easier than deriving an analytic expression of the environment directly.

- *Reinforcement learning (RL)*: Different from DP, the RL algorithms are used when the environment is too complex to construct a model for it. RL can be seen as model-free, sample-based DP, and DP can be seen as model-base RL. Since constructing a model is expensive or difficult in some cases, RL algorithms learn the environment with the limited data which can be obtained by simulations.

Both DP and RL aim to choose the optimal strategy that maximizes the cumulative rewards over a long period of time for MDP problems. More precisely, an MDP consists three signals: a *state*, an *action*, and a *reward*. The state describes the state of the decision maker at each time step. The action describes the chosen decision which affects the environment. The reward is an evaluation signal of the chosen action which is provided by the environment. The goal is to learn the optimal *policy*, which leads the optimal actions for every state. An illustration of the MDP framework is shown in Figure 1.1. In a more formal way, an MDP is specified by four components $\{S, A, p, r\}$, where

- S is the state space, which contains all states $\boldsymbol{s}$. S could be either discrete or continuous.

- A is the action space, which contains all possible actions $\boldsymbol{a}$. A could be either discrete or continuous.

- p could be either the transition probability or transition probability density, while the state space and action space are discrete or continuous. $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$

yields the conditional probability of next state $\boldsymbol{s}'$ given current state $\boldsymbol{s}$ and action $\boldsymbol{a}$ to be taken.

- $r(\boldsymbol{s}, \boldsymbol{a})$ is the reward function of the state $\boldsymbol{s}$ and the action $\boldsymbol{a}$.

An MDP algorithm can produce two kinds of policies: deterministic policies $\pi(\boldsymbol{s})$, which always choose the same action $\boldsymbol{a} = \pi(\boldsymbol{s})$ for a given state $\boldsymbol{s}$, and stochastic policies $\pi(\boldsymbol{a}|\boldsymbol{s})$, which is the conditional probability (density) of action $\boldsymbol{a}$ to be taken given state $\boldsymbol{s}$. At each time step t, the decision maker observes its current state $\boldsymbol{s}_t$ and chooses an action according to $\pi(\boldsymbol{s}_t)$ or $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$. Then the decision maker transits to the next state $\boldsymbol{s}_{t+1}$ following the transition $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. At the same time, a reward $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is assigned and revealed to the decision maker. The number of the time steps could be infinite or finite in reality, which are called infinite horizon MDPs and finite horizon MDPs, respectively. Moreover, there are two kinds of evaluation functions (i.e., the return) for infinite and finite horizon MDPs as follows.

For the finite horizon MDP, the return is the sum of rewards over the entire time horizon:

$$R_\pi(T) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \tag{1.1}$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expectation taken over the trajectory $\{\boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{s}_T, \boldsymbol{a}_T\}$ generated by the policy π, T is the length of the time horizon. It is clear the above definition of the return is meaningless for infinite time horizon MDPs, since the sum of rewards becomes unbounded in this case. Therefore, the discounted return is defined instead which is bounded by $\frac{\|r(\cdot,\cdot)\|_\infty}{1-\alpha}$:

$$R^\alpha(\pi) = \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \alpha^{t-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right],$$

where $\alpha \in [0, 1)$ is the discount factor. Similarly, the average return is defined as

$$\rho_r(\pi) = \lim_{T \to \infty} \mathbb{E}_\pi \left[\frac{1}{T} \sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right].$$

Considering the state structure of MDPs, the returns defined in above equations are not enough to evaluate the performance of a policy in a given state $\boldsymbol{s}$.

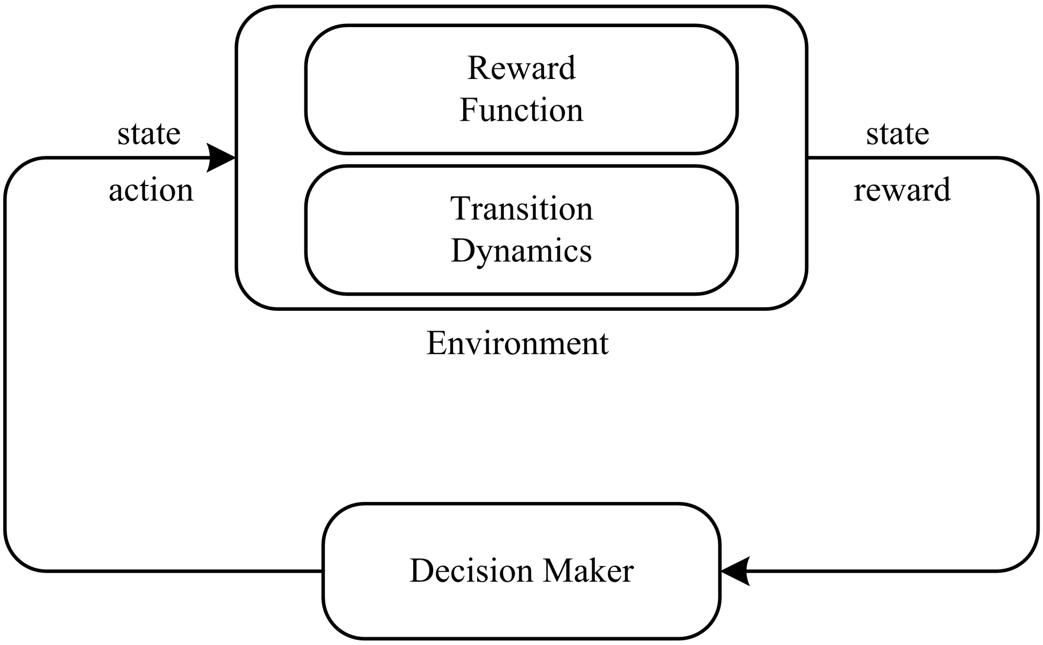


Figure 1.1: The illustration of the interaction between the decision maker and the environment in MDPs.

Another category of evaluations is essential, which takes the state and action structures into account. For finite horizon MDPs, we could define the value function $\mathcal{V}_r^\pi(\boldsymbol{s}, T)$ and the state-action value function $Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, T)$, which are efficient tools for learning the policy. The value function $\mathcal{V}_r^\pi(\boldsymbol{s}, T)$ is used to indicate the performance of policy π starting from a given state $\boldsymbol{s}$, which is defined as

$$\mathcal{V}_r^\pi(\boldsymbol{s}, T) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_1 = \boldsymbol{s} \right].$$

Similarly, the state-action value function $Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, T)$ is defined for evaluating policy π starting form a pair of given state $\boldsymbol{s}$ and action $\boldsymbol{a}$ which is defined as

$$Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, T) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_1 = \boldsymbol{s}, \boldsymbol{a}_1 = \boldsymbol{a} \right].$$

To the purpose of boundness, the discount factor and the average criteria are used for infinite horizon MDPs. The value function $\mathcal{V}_r^\pi(\boldsymbol{s})$ and the discounted value function $\mathcal{V}_r^\pi(\boldsymbol{s}, \alpha)$ are defined as

$$\mathcal{V}_r^\pi(\boldsymbol{s}) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \left(r(\boldsymbol{s}_t, \boldsymbol{a}_t) - \rho_r(\pi) \right) | \boldsymbol{s}_1 = \boldsymbol{s} \right],$$

$$\mathcal{V}_r^\pi(\boldsymbol{s}, \alpha) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \alpha^{t-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_1 = \boldsymbol{s} \right].$$

The state-action value function $Q_r^\pi(\boldsymbol{s}, \boldsymbol{a})$ and the discounted state-action value function $Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, \alpha)$ for infinite horizon MDPs are defined as

$$Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \left(r(\boldsymbol{s}_t, \boldsymbol{a}_t) - \rho_r(\pi) \right) | \boldsymbol{s}_1 = \boldsymbol{s}, \boldsymbol{a}_1 = \boldsymbol{a} \right],$$

$$Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, \alpha) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \alpha^{t-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_1 = \boldsymbol{s}, \boldsymbol{a}_1 = \boldsymbol{a} \right].$$

The famous Bellman equation (Bellman, 1957) shows the relationship of the above evaluation criteria:

$$\begin{aligned}
\mathcal{V}_r^\pi(\boldsymbol{s}, T) =& \mathbb{E}_{\boldsymbol{a} \sim \pi}[Q_r^\pi(\boldsymbol{s}, \boldsymbol{a}, T)] \\
=& \mathbb{E}_{\boldsymbol{a} \sim \pi} \left[r(\boldsymbol{s}, \boldsymbol{a}) + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) \mathcal{V}_r^\pi(\boldsymbol{s}', T) \right],
\end{aligned}$$

where $\mathbb{E}_{\boldsymbol{a}\sim\pi}[\cdot]$ denotes the expectation taken over the action space. Similarly, the Bellman equation for infinite horizon MDPs indicates:

$$\begin{aligned}\mathcal{V}_r^\pi(\boldsymbol{s}) =&\mathbb{E}_{\boldsymbol{a}\sim\pi}[Q_r^\pi(\boldsymbol{s},\boldsymbol{a})]\\ =&\mathbb{E}_{\boldsymbol{a}\sim\pi}\left[r(\boldsymbol{s},\boldsymbol{a})-\rho_r(\pi)+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^\pi(\boldsymbol{s}')\right],\end{aligned}$$

and

$$\begin{aligned}\mathcal{V}_r^\pi(\boldsymbol{s},\alpha) =&\mathbb{E}_{\boldsymbol{a}\sim\pi}[Q_r^\pi(\boldsymbol{s},\boldsymbol{a},\alpha)]\\ =&\mathbb{E}_{\boldsymbol{a}\sim\pi}\left[r(\boldsymbol{s},\boldsymbol{a})+\alpha\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^\pi(\boldsymbol{s}',\alpha)\right].\end{aligned}$$

Several kind of efficient algorithms are proposed for optimizing the above criteria using the Bellman equation, i.e., the policy iteration, value iteration and policy search algorithms:

- *Value iteration* algorithms learn the optimal policy by searching for the optimal value function. After enough number of iterations, the optimal value function can be learned from the collected data, which yields the optimal policy directly.

- *Policy iteration* algorithms evaluate policies by constructing their value functions. At the same time, the policy is improved by the these value functions. After enough number of iterations, the improved policies converge to the optimal policy.

- *Policy search* algorithms directly solve the optimization problems which maximize the above criterion, e.g., by gradient ascent.

In this thesis, we mainly focus on developing the last two kinds of algorithms since a decision should be made at each time step in the online decision making problems.

1.3 Online Markov Decision Processes

Learning in non-stationary environments gains extensive attention, since environments are not always fixed in reality. The tasks are more challenging due to the lack of information about the future evolution of environments. A plenty of problems concern how to make decisions with this challenging non-stationary environment, e.g., the multi-armed bandit problem, the competition game problem, and the online control problem. In this section, we consider MDPs with changing environments called online MDPs which is a promising generalization of standard MDPs. A typical framework of the online MDP is shown in Figure 1.2. The decision maker sequentially chooses an action after observing its current state. The environment shows the reward of the chosen action to the agent, which is assumed to be partially affected by some unknown changing factors abruptly. Overall, an online MDP problem is an extension of both online decision making and reinforcement learning (Yu et al., 2009):

- In an *online decision making* problem, the agent needs to make a decision at each time step without the knowledge about the future environment (Kalai and Vempala, 2005). A certain cost function will be observed only after the decision is made at each time step, and the goal is to minimize the regret against the best single decision. There is no assumption on the dynamics in the online decision making problem, and thus the decision can switch from one to another abruptly.

- In *reinforcement learning*, the dynamics are assumed to be Markovian. The reward function and transition dynamics are fixed but unknown to the agent, and thus the estimated reward function and transition function will converge to the true ones if sufficient number of samples are observed. The goal is to find the optimal policy which maximizes the cumulative reward without full information about the environment.

In a more formal way, an online MDP is specified by $\{S, A, p, [r_t]_{t=1,\dots,\infty}\}$, where

- S is the state space, which could be either discrete or continuous.

- A is the action space, which could be either discrete or continuous.

- $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) : S \times S \times A \to \mathbb{R}$ is the transition probability (density).

- $r_1, r_2, \dots$ is an infinite reward function sequence, only $r_1, \dots, r_{t-1}$ are observed at time step t.

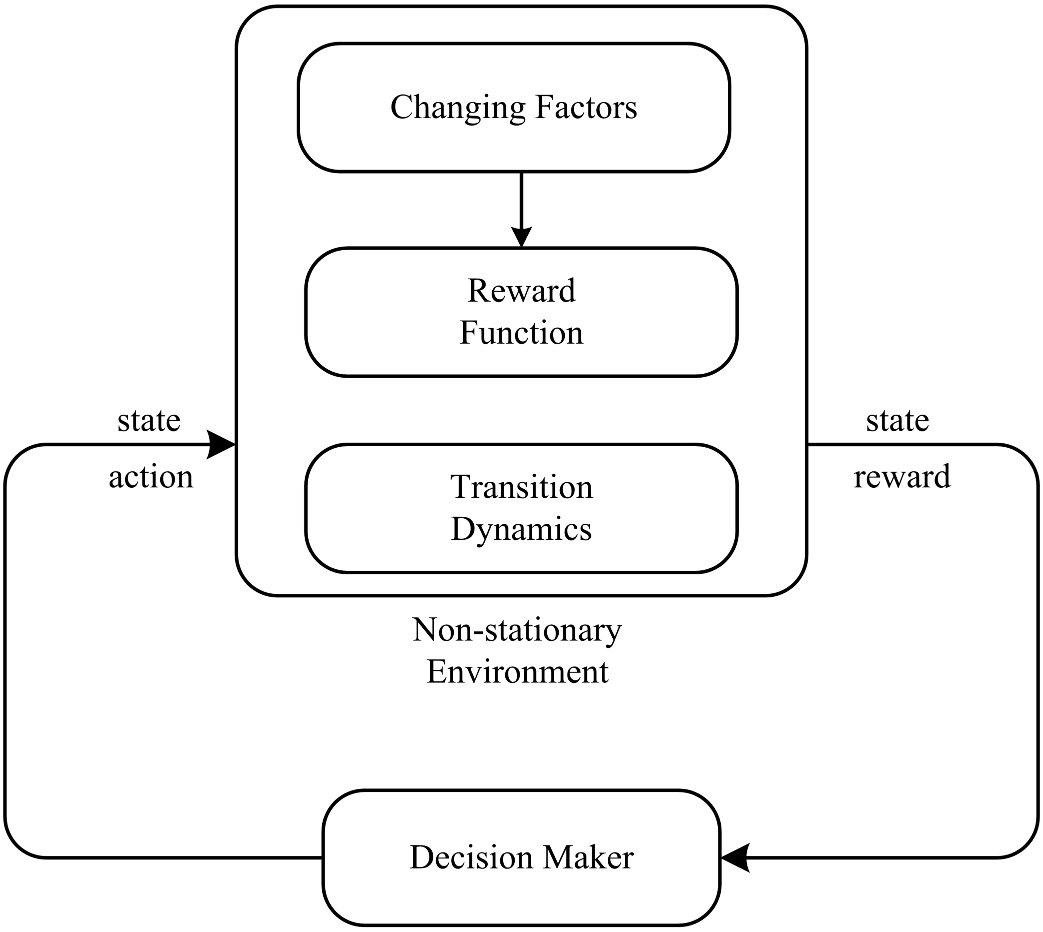


Figure 1.2: The illustration of the interaction between the decision maker and the environment in online MDPs.

The objective of an online MDP algorithm is to produce a strategy of choosing an action at each time step. Similarly to the MDP, the time-dependent policy π_t here could be either deterministic or stochastic. A time-dependent deterministic policy always provides the same action $\boldsymbol{a} = \pi_t(\boldsymbol{s})$ for a given state $\boldsymbol{s}$ at each time step t. On the other hand, a time-dependent stochastic policy $\pi_t(\boldsymbol{a}|\boldsymbol{s})$ yields the conditional probability (density) of action $\boldsymbol{a}$ to be taken at state $\boldsymbol{s}$ at each time step t. A formal definition of the return for an online MDP algorithm $\mathcal{A}$ over T time steps is given as

$$R_{\mathcal{A}}(T) = \mathbb{E}_{\{\pi_1,\ldots,\pi_T\}}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right],$$

where $\{\pi_1, \ldots, \pi_T\}$ is the policy sequence (i.e., the time-dependent policy) provided by $\mathcal{A}$. The expectation $\mathbb{E}_{\{\pi_1,\ldots,\pi_T\}}[\cdot]$ denotes the expectation taken over the trajectory $\{\boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{s}_T, \boldsymbol{a}_T\}$ following the algorithm $\mathcal{A}$. Similarly, we define the return and the average return of a fixed policy π over T time steps as

$$R_{\pi}(T) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}, \boldsymbol{a})\right]. \tag{1.2}$$

Note that the above definition is similar to the definition of the return of standard MDPs in Eq.(1.1). We use the same notation since the only difference is the reward function is changing. In the rest of this thesis, we consider the definition in Eq.(1.2).

As we showed in Section 1.2, the MDP is a powerful framework to solve stochastic optimization problems (Parr, 1998). Similarly, the online MDP could be applied to many optimization problems from robotics to finance with non-stationary environments. Several examples are provided as follows:

Tracking the moving target: Consider the problem of tracking a moving target where the goal is to minimize the distances of an artificial agent and a target agent over some period of time T. If the target agent moves arbitrarily (adversarially), the artificial agent should adapt the moving strategy which could minimize the distances. The distances are partially influenced by the target agent, and partially influenced by the agent. As shown in Abbasi-Yadkori et al. (2013), the problem could be solved by dynamic programming if the trajectory of the target is known in advance. However, the trajectory is usually unknown and the prior knowledge might not be trustable. In this scenario, let us assume the state $\boldsymbol{s}_t$ is the current position of the agent which can be observed. At each time step t, the action $\boldsymbol{a}_t$ is decided which is the moving distance and direction of the agent. The next state $\boldsymbol{s}_{t+1}$ is captured by the Markovian dynamic (i.e., transition). The distance evaluation r_t depends on the position of the moving target at each time step, which is abruptly changing. The online MDP model is capable of solving this problem and obtains an effective tracking strategy even if the trajectory is unknown.

Inventory control: Consider the problem of how to make an ordering list according to the amount of items the store holds. Since the demand distribution of the customers could change from time to time, the manager of the store should change the ordering strategy correspondingly. However, the behaviors of the customers are usually unpredictable which makes the problem harder to control. We assume that the current number of the items in the storage is the state $\boldsymbol{s}_t$. At each time step t, the manager needs to decide how many items to order from the supplier, which is the action $\boldsymbol{a}_t$. The next state $\boldsymbol{s}_{t+1}$ is decided by a linear function in this case. The goal is to maximize the cumulative revenue (i.e., reward function), which is partly decided by the demand distribution of the customers. Since the reward function is changing, this problem can be formulated as an online MDP.

Recommender System: It concerns the problem of providing recommendations to the customers depending on the users' profiles. The goal is to maximize the users' acceptance of the recommendations. Since there are a large number of users available over some time period, the users' profiles are usually switching frequently. Let us assume that the state $\boldsymbol{s}_t$ is the current page that the user is viewing. According to a recommendation strategy, the recommender system offers a page $\boldsymbol{a}_t$ according to the observations. The reward function r_t is not only decided by the offered page at each time step but also decided by the user. This problem can be formulated as an online MDP problem with a layer-structured state space (e.g., online episodic MDP) (Neu et al., 2010a; Zimin and Neu, 2013).

1.4 Regret

As mentioned beforehand, we show that the online decision making problem is proposed for choosing the best decision sequence which maximize the outcomes over a period of time. Moreover, we would expect the cumulative outcomes of our online decision making algorithm are independent of the length of the time horizon. More realistically, there is no hope of comparing the performance of an online decision making algorithm to the performance of the best strategy sequence. Therefore, we would like to introduce the notion of *regret* which is a relative performance evaluation with respect to the best fixed strategy. The motivation (Nisan et al., 2007) could be viewed as: Our algorithm adapts the strategy based on the observations and receives the outcomes. We would like to avoid the embarrassment that we could receive much more outcomes if we used a simple fixed strategy all the time.

As mentioned in Section 1.2, the performance is evaluated by the return over some time period. Ideally, the objective is to generate a optimal time-dependent policy which maximizes the cumulative rewards over T time steps. Given the fact online MDP problems are also online decision making problems, we would like to define the regret against the best policy which is a relative performance evaluation with respect to the best fixed policy. We treat the best fixed policy as a baseline since the best time-dependent policy is simply too strong as a baseline. That is, the preferred baseline will be $\sup_{\pi\in\Pi} R_\pi(T)$ rather than $\sup_{\{\pi_1,\ldots,\pi_T\}\in\Pi^T} \mathbb{E}_{\{\pi_1,\ldots,\pi_T\}}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$. A formal definition of the regret for an online MDP algorithm $\mathcal{A}$ over T time steps can be written as

$$L_{\mathcal{A}}(T) = \sup_{\pi\in\Pi} R_\pi(T) - R_{\mathcal{A}}(T). \tag{1.3}$$

Therefore, we are interested in constructing algorithms that minimize the regret instead of maximizing the return. Furthermore, an online MDP algorithm $\mathcal{A}$ has no regret, if for any arbitrary reward function sequence, the regret (1.3) is $o(T)$ (i.e., sublinear regret),

1.5 Contributions

This thesis contributes to the theory of online decision making problems in non-stationary Markovian environments. The state-of-the-art work (Even-Dar et al., 2003, 2009; Yu et al., 2009; Dick et al., 2014) is not extensible to continuous state space online MDP problems without additional assumptions and theorems since a strategy of choosing actions is learned for each state individually. Although one typical way is to combine these algorithms with the discretization of continuous state spaces, it also suffers from the *curse of dimensionality* (Bellman, 1957), which means that the time and space requirements to solve an MDP is exponentially increased as the dimension increases. The online MDP problem with a large discrete state space also suffers from the same issue as the continuous problem. For this purpose, two algorithms are proposed, which are aimed at solving online MDP problems with a large or continuous state space. The first proposed algorithm settles the continuous problem by parameterizing the policy space. Under a concavity assumption, the first proposed algorithm is proved to perform asymptotically equal to the best fixed policy. Furthermore, the second proposed algorithm parameterizes the value function which leads us to the policy directly. Without the concavity assumption, the second proposed algorithm is computationally more efficient and extensible to continuous problems in exchange for large regret. The illustration of the connection between existing work and the proposed algorithms is shown in Figure 1.3. Through regret analysis, we illustrate the proposed algorithms are no-regret algorithms with different regret bounds as shown in Table 1.1.

- Chapter 3 considers online MDP problems with continuous state and action spaces. A policy search type method is proposed and shown to achieve a sublinear regret in this case.

- Chapter 4 introduces the proposed policy iteration type method for online MDP problems with a large scale state space.

In this section, we briefly present the main results of the above researches.

Table 1.1: Regret bounds of proposed algorithms in this thesis.

	Full Information	Bandit Feedback
OPG	Concavity $O(\sqrt{T})$ Strong Concavity $O(\log T)$	Concavity $O(\sqrt{T})$
OMDP-PI	$o(T)$	Future Work

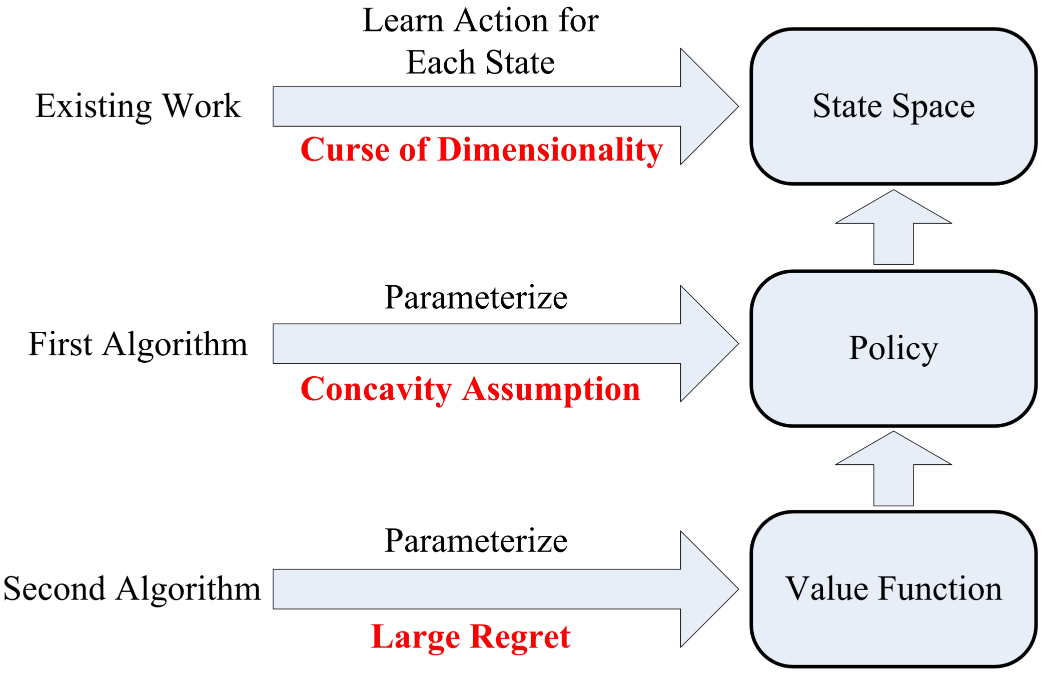


Figure 1.3: Connection between existing work and the proposed algorithms.

1.5.1 Continuous State and Action Spaces Online MDPs

As we have shown in Section 1.3, plenty of real online optimization problems can be formulated as online MDPs, e.g. tracking the moving target, inventory control, and recommender system. By the development of this model, the demand of solving large scale online MDP problems has been arisen in reality. Especially, the online MDP problem with continuous state and action spaces is a challenging research direction since we can not perform greedy methods searching for the best policy. In recent years, some online MDP algorithms have been proposed with two kinds of main ideas: expert algorithm based methods (Even-Dar et al., 2003, 2009; Yu et al., 2009) and online linear optimization based algorithms (Dick et al., 2014). However, these algorithms are not extensible to the continuous setting without additional assumptions and theorems since they learn the action distribution for each state individually. In Chapter 3, we settle this problem by a policy search type method. The proposed online policy gradient (OPG) algorithm solves the continuous problem with a parameterized policy model. To the best of our knowledge, this is the first work to give an online MDP algorithm that can handle continuous state and action spaces with guarantee. Regarding different scenarios, we provide several main theorems as follows:

- Full information feedback with a concavity assumption: Suppose that at each time step t, the entire reward function $r_t(\boldsymbol{s}, \boldsymbol{a})$ is revealed to the algorithm (full information feedback). We assume that $\rho_{r_t}, \forall t = 1, \ldots T$ are concave with respect to the policy parameter. In this scenario, the regret against the best fixed policy of the OPG algorithm is $O(\sqrt{T})$.

- Bandit feedback with a concavity assumption: Suppose that at each time step t, only the reward value $r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is revealed to the algorithm (bandit feedback). We assume that $\rho_{r_t}, \forall t = 1, \ldots T$ are concave with respect to the policy parameter. In this scenario, the regret against the best fixed policy of the OPG algorithm is $O(\sqrt{T})$.

- Full information feedback with a strong concavity assumption: We assume that $\rho_{r_t}, \forall t = 1, \ldots T$ are strong concave with respect to the policy parameter. In this scenario, the regret against the best fixed policy of the OPG algorithm with full information feedback is $O(\log T)$.

Through theoretical results, we show the OPG algorithm is a no-regret algorithm in different scenarios. Furthermore, we demonstrate the experimental behavior of the OPG algorithm, which substantiates the theoretical results.

1.5.2 Large State Space Online MDPs

The online MDP problem could solve many real problems from robotics to finance with time-varying environments. It is natural to work on the large (possibly infinite) state space problems, such that we consider to propose an algorithm with less computation complexity in exchange for large regret.

The proposed algorithm OMDP-PI online Markov decision processes with policy iteration (OMDP-PI) is motivated by the idea of combining the function approximation with policy iteration. With full information of reward functions, the proposed OMDP-PI algorithm is proved to achieve following results:

- The regret against the best policy of the OMDP-PI algorithm is sublinear.
- At each time step, the update rule of the proposed OMDP-PI algorithm could be performed in $O(|S|^{2.3728639} + |S|^2|A|)$, which is more efficient than existing methods.
- The linear approximation can be used together with the OMDP-PI algorithm for large (infinite) state space problems, where the convergence is guaranteed.
- The OMDP-PI algorithm could be extended to a more general algorithm called the Online Markov Decision Processes with Stochastic Iteration (OMDP-SI) algorithm. Under some additional assumptions, the OMDP-SI algorithm achieves a sublinear regret as well.

Through a grid world experiment, we illustrate the experimental performance of the OMDP-PI algorithm, which verifies the theoretical regret analysis.

1.6 Outline of This Thesis

This thesis consists of five chapters as shown in Figure 1.4. In this section, we present the organization of this thesis.

In Chapter 2, we analyze some related state-of-the-art algorithms for discrete state and action spaces online MDP problems. Two types of fundamental techniques: expert algorithms and online convex optimization are introduced and reviewed in Section 2.1.1 and Section 2.2.1, respectively. Then existing online MDP algorithms with finite state and action spaces are analyzed, e.g., online MDPs with experts in Section 2.1.2, online MDPs with online linear optimization in Section 2.2.2.

In Chapter 3, we propose an online policy gradient algorithm for online MDP problems with continuous state and action spaces. In Section 3.1, we show the motivation and the background knowledge of the continuous problem. In Section 3.2, we present the proposed algorithm in detail. In Section 3.3 and Section 3.4, we present the main theorems of the proposed algorithm with different feedback under different assumptions. Section 3.3 describes the theoretical results of the OPG algorithm with full feedback under concavity and strong concavity assumptions. Section 3.4 describes the theoretical result with bandit feedback under a concavity assumption. In Section 3.5, we demonstrate the experimental behavior of the proposed algorithm with two toy experiments. Finally, the proofs of all the theoretical results are presented in Section 3.6.

In Chapter 4, we propose an online MDPs with a policy iteration algorithm for large state space online MDP problems. In Section 4.1, we present some necessary preliminaries. Section 4.2 describes the proposed algorithm and the main theorems in detail. An extension of the OMDP-PI algorithm with function approximation for large state space problems is also provided in Section 4.2. In Section 4.3, we further extend the proposed algorithm to a more general online MDPs with stochastic iteration algorithm. Through a grid world experiment, we illustrate the experimental performance of the OMDP-PI algorithm in Section 4.4. In Section 4.5, we present a comparison of the proposed algorithm with previous works. In Section 4.6, the proofs of all the theoretical results are presented.

In Chapter 5, we present the conclusions and future work.

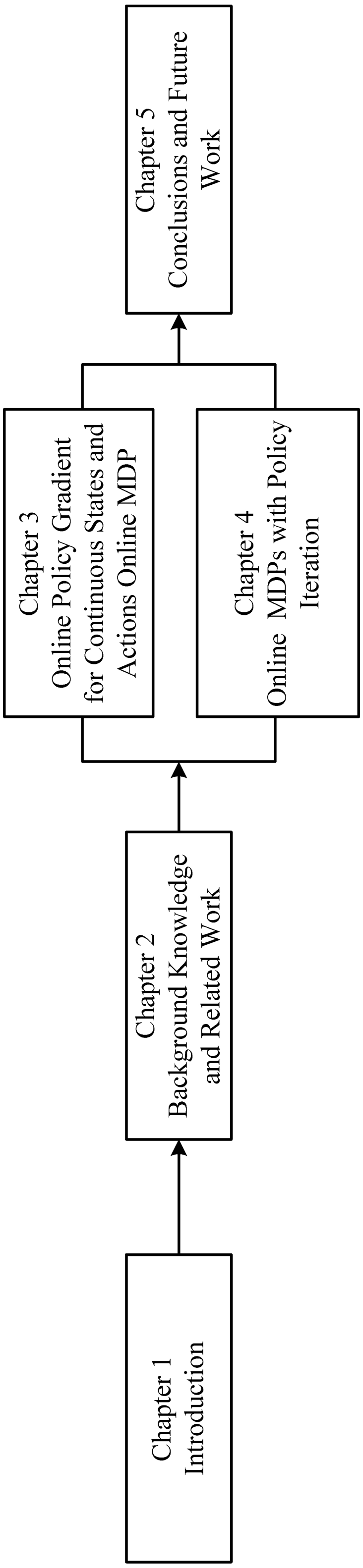

Chapter 1
Introduction
Chapter 2
Background Knowledge and Related Work
Chapter 3
Online Policy Gradient for Continuous States and Actions Online MDP
Chapter 4
Online MDPs with Policy Iteration
Chapter 5
Conclusions and Future Work

Figure 1.4: Organization of this thesis

Chapter 2

Background Knowledge and Related Work

In this chapter, we introduce the related work for discrete state space online MDP problems. First of all, we introduce the online MDPs with experts algorithm in Section 2.1. A fundamental online decision making problem called expert's prediction is presented in Section 2.1.1, which is the basic technique involved in the online MDPs with experts algorithm. Then we present the main algorithm and the theoretical results in Section 2.1.2 and Section 2.1.3. In Section 2.2, we introduce the online MDPs with online linear optimization algorithm. The fundamental online convex optimization problem is presented in Section 2.2.1. The theoretical results are presented in Section 2.2.2.

2.1 Online MDPs with Experts

The *online MDPs with Experts* algorithm (Even-Dar et al., 2003, 2009) aims to investigate how to incorporate the idea of experts to the MDP structure. A review of the expert algorithms is presented in the beginning of this section. Then we introduce two efficient ways to further extend expert algorithms for solving online MDP problems. At last, we introduce the theoretical results for these two approaches.

2.1.1 Online Decision Making with Expert Algorithm

An extensively studied online decision making problem is the expert prediction, whose goal is to predict the best decision by choosing the best expert's prediction (Littlestone and Warmuth, 1994; Vovk, 1990; Cesa-Bianchi and Lugosi, 2006). More precisely, the decision maker chooses its own prediction y_t from the experts' prediction set $D = \{y_{e,t} : e \in E\}$ at each time step t, where E is the set of experts indexed by e. The expert prediction follows the same protocol as the repeated game:

- For each time step $t = 1, 2, \ldots$
    1. Observe current experts' predictions $y_{e,t}, \forall e \in E$;
    2. Choose the prediction $y_t \in D$;
    3. Observe the real outcome y_t^* from the environment;
    4. Update the strategy of choosing the expert by the loss $l(y_t, y_t^*)$;

The objective of an expert prediction algorithm is to minimize the cumulative regret, the difference of the cumulative losses with respect to each expert over T time steps. The regret is defined as

$$L_{\mathcal{A}}(T) = \sup_{e \in E} \sum_{t=1}^{T} \left(l(y_t, y_t^*) - l(y_{e,t}, y_t^*) \right).$$

By the randomness of the decision strategy, we divide the expert prediction algorithms into deterministic and stochastic algorithms. Here, we introduce one deterministic and two stochastic experts algorithms in this sections.

The *weighted majority* (WM) algorithm (Littlestone and Warmuth, 1994) makes the decision by weighted voting, where the weights are determined by the mistakes each expert made. At each time step the weights of incorrect experts will be reduced by the penalty factor $\beta \in [0, 1)$ defined in advance. The WM algorithm is summarized in Figure 2.1.

Now we move to the analysis of the WM algorithm. Any deterministic algorithm (e.g., WM algorithm) cannot achieve a sub-linear regret in general, since that the environment can always choose an adversarial feedback, which makes the algorithm suffers 1 loss at each time step. However, the number of mistakes made by the WM algorithm can be bounded by the following theorem.

Theorem 2.1. *Let $M_{WM}(T)$ and $M^*(T)$ be the number of mistakes made by the WM algorithm and the best expert $e^* \in E$ over T time steps. Then $M_{WM}(T)$ can be bounded as*

$$M_{\mathrm{WM}}(T) \leq \frac{\log |E| + M^*(T) \log 1/\beta}{\log 2/(1+\beta)},$$

where $|E|$ denotes the cardinality of the expert set.

Proof. The proof we present follows Mohri et al. (2012). To obtain the result in Theorem 2.1, we firstly introduce the potential function defined as

$$W_t = \sum_{e \in E} w_{e,t}.$$

Then the update rule leads us to the following result:

$$W_{t+1} \leq \frac{1+\beta}{2} W_t, \quad if \; y_t \neq y_t^*.$$

This result can be obtain by the fact that $\sum_{e:f_{e,t} \neq y_t^*} w_{e,t} \geq \sum_{e:f_{e,t}=y_t^*} w_{e,t}$ when $y_t \neq y_t^*$. Clearly, W_t is always non-negative for all $t = 1, 2, \ldots, T$. Therefore we have $W_T \geq w_{e^*,T} = w_{e^*,1} \beta^{M_T^*}$, which yields the following inequalities:

$$\beta^{M^*(T)} \leq W_T \leq W_1 \left(\frac{1+\beta}{2} \right)^{M_{WM}(T)}.$$

By taking the logarithm, we conclude the proof. □

The mistake bound in Theorem 2.1 substantiates that the deterministic algorithm cannot achieve a sublinear regret. A stochastic variant of the WM algorithm is the the *randomized weighted majority* (RWM) algorithm (Littlestone and Warmuth, 1994), which overcomes the linear regret issue of deterministic algorithms. The basic idea is to predict y_t by following the expert e with probability $w_{e,t} / \sum_{e \in E} w_{e,t}$ at time step t, as shown in Figure 2.2. The regret of the RWM algorithm is bounded by the following theorem:

Theorem 2.2. *Let $\mathcal{L}_{\mathrm{RWM}}(T)$ and $\mathcal{L}_{e^*}(T)$ be the expected losses of the RWM algorithm and the best expert $e^* \in E$ over T time steps. Then the regret $L_{\mathrm{RWM}}(T) = \mathcal{L}_{RWM}(T) - \mathcal{L}_{e^*}(T)$ satisfies*

$$L_{\mathrm{RWM}}(T) \leq 2\sqrt{T \log |E|},$$

by setting $\beta = \max\{1/2, 1 - \sqrt{\frac{\log |E|}{T}}\}$.

```
Initialize the weights for all $e \in E$ with $w_{e,1} = 1$.
for $t = 1, \ldots, \infty$ do
    Observe experts' predictions $y_{e,t}, \forall e \in E$.
    Choose the prediction by
        $y_t = \begin{cases} 1, & if\ \frac{\sum_{e \in E} w_{e,t} y_{e,t}}{\sum_{e \in E} w_{e,t}} \geq 1/2 \\ 0, & otherwise \end{cases}$
    Observe $y_t^*$.
    if $y_t \neq y_t^*$ then
        Update $w_{e,t},\ \forall e \in E$ according to
            $w_{e,t+1} = \begin{cases} \beta w_{e,t} & if\ y_{e,t} \neq y_t^* \\ w_{e,t} & otherwise \end{cases}$
    end if
end for
```

Figure 2.1: Weighted majority algorithm

Proof. The proof we present follows Mohri et al. (2012). Here we use the same potential functions $W_t = \sum_{e \in E} w_{e,t}, \forall t = 1, \ldots, T$. We can derive the upper bound for W_{t+1} by the update rule as

$$\begin{aligned} W_{t+1} = & \sum_{e: f_{e,t} = y_t^*} w_{e,t} + \beta \sum_{e: f_{e,t} \neq y_t^*} w_{e,t} \\ = & W_t + (\beta - 1) \sum_{e: f_{e,t} \neq y_t^*} w_{e,t} \\ = & W_t + (\beta - 1) W_t l_{\text{RWM}}(t) \\ = & (1 + (\beta - 1) l_{\text{RWM}}(t)) W_t \\ = & W_1 \prod_{i=1}^{t} (1 + (\beta - 1) l_{\text{RWM}}(i)), \end{aligned}$$

where $l_{\text{RWM}}(i)$ is the expected loss of the RWM algorithm at time step i. Combining the upper bound with the lower bound for W_{T+1} as

$$\beta^{\mathcal{L}_{e^*}(T)} = w_{e^*, T+1} \leq W_{T+1} = |E| \prod_{i=1}^{T} (1 + (\beta - 1) l_{\text{RWM}}(i)).$$

By taking the logarithm, we can achieve the following bound:

$$\begin{aligned} \mathcal{L}_{e^*}(T) \log \beta \leq & \log |E| + \sum_{i=1}^{T} \log (1 - (1 - \beta) l_{\text{RWM}}(i)) \\ = & \log |E| - (1 - \beta) L_{\text{RWM}}(T), \end{aligned}$$

where the last inequality can be obtained by using the fact $\log (1 - x) \leq -x, \forall x < 1$. Such that $\mathcal{L}_{\text{RWM}}(T)$ and $\mathcal{L}_{e^*}(T)$ satisfy the following inequalities

$$\begin{aligned} \mathcal{L}_{\text{RWM}}(T) \leq & \frac{\log (1 - (1 - \beta))}{1 - \beta} \mathcal{L}_{e^*}(T) + \frac{\log |E|}{1 - \beta} \\ \leq & (1 - \beta) T + \mathcal{L}_{e^*}(T) + \frac{\log |E|}{1 - \beta}, \end{aligned}$$

where the last inequality comes from the inequality $-\log (1 - x) \leq x + x^2, \forall x \in [1, 1/2]$. Thus, by setting $\beta = \max \{1/2, 1 - \sqrt{\frac{\log N}{T}}\}$ we conclude the proof. □

Although both the WM and RWM algorithms considered the binary prediction with the 0-1 loss function, it is also possible to extend them with another loss function.

```
Initialize the weights for all $e \in E$ with $w_{e,1} = 1$, $p_{e,1} = 1/|E|$ for all $e \in E$.
for $t = 1, \ldots, \infty$ do
    Observe experts' predictions $y_{e,t}, \forall e \in E$.
    Choose the prediction $y_t = y_{e,t}$ with probability $p_{e,t}$
    Observe $y_t^*$.
    if $y_t \neq y_t^*$ then
        Update $w_{e,t},\ \forall e \in E$ according to
            $w_{e,t+1} = \begin{cases} \beta w_{e,t} & if\ y_{e,t} \neq y_t^* \\ w_{e,t} & otherwise \end{cases}$
    end if
    Update $p_{e,t+1} = \frac{w_{e,t+1}}{\sum_{e \in E} w_{e,t+1}}$ for all $e \in E$.
end for
```

Figure 2.2: Randomized weighted majority algorithm

Now we will present another stochastic algorithm called *follow the perturbed leader* (FPL) algorithm (Cesa-Bianchi and Lugosi, 2006), which suggests that the decision maker should follow the expert who performed best over previous trials. As we claimed before, any deterministic algorithm cannot achieve a sublinear regret in general. To randomized the strategy, a simple idea is to introduce a random perturbation variable which leads the decision maker following the perturbed leader instead of the real leader. Before introducing the algorithm, let us define the perturbation random vectors as $\boldsymbol{n}_t, \forall t = 1, \ldots, \infty$. Thus, $\boldsymbol{n}_t$ is a $|E|$-dimensional vector with $|E|$ random perturbation variables $n_{e,t}, e \in E$ for $|E|$ experts. At each time step t, the FPL algorithm chooses the prediction as $y_t = f_{I_t,t}$ where

$$I_t = \mathrm{argmin}_{e \in E} \left(\sum_{i=1}^{t-1} l_{e,i} + n_{e,t} \right).$$

In the above equation, we use $l_{e,i}$ instead of $l(y_{e,i}, y_i^*)$ for notational simplicity. Clearly, the regret of the FPL algorithm depends on the perturbation random vectors $\boldsymbol{n}_t, t = 1, \ldots, \infty$. The analysis of the FPL algorithm with a uniform random vector is presented in Theorem 2.3.

Theorem 2.3. *After T time steps, the regret against the best expert of the FPL algorithm with uniform perturbation random variables on $[0, \epsilon]$ satisfies*

$$L_{\mathrm{FPL}}(T) \leq \epsilon + \frac{T|E|}{\epsilon}.$$

Proof. The proof we present follows Cesa-Bianchi and Lugosi (2006), where the main idea of the proof is to consider the predictor who looks one step ahead as

$$\hat{I}_t = \mathrm{argmin}_{e \in E} \left(\sum_{i=1}^{t} l_{e,i} + n_{e,t} \right).$$

Note that $\hat{I}_t$ is not a real predictor, since $l_{e,t}, \forall e \in E$ is not available at time step t. However, we treat $\hat{I}_t$ as a fictitious predictor which does not perform much better than I_t. Thus, the bound for the difference between $\hat{I}_t$ and the best expert e^* can be obtain by following inequalities.

$$\begin{aligned}
&\sum_{i=1}^{t}\Big(l(\hat{I}_i,i)+n_{\hat{I}_i,i}-n_{\hat{I}_i,i}\Big)\\
&\leq \min_{e\in E}\sum_{i=1}^{t}\left(l(e,i)+n_{e,i}-n_{e,i-1}\right)\\
&\leq \min_{e\in E}\sum_{i=1}^{t} l(e,i)+n_{e^*,t}. \qquad (2.1)
\end{aligned}$$

The perturbation random variables $n_{e,t}, \forall e \in E, t = 1, \ldots, T$ are considered as uniform random variables in Hannan (1957), which is defined as

$$f(n_{e,t}) = \begin{cases} \frac{1}{\epsilon} & n_{e,t} \in [0,\epsilon],\\ 0 & otherwise. \end{cases}$$

Then taking expectation over Eq.(2.1) as

$$\mathbb{E}\sum_{i=1}^{t} l(\hat{I}_i,i) \leq \min_{e\in E}\sum_{i=1}^{t} l(e,i) + \mathbb{E}\max_{e\in E} n_{e,t}.$$

By defining $F_t(\boldsymbol{n}) = l(I_t, y_t^*)$, the fictitious predictor satisfies the following equation:

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T} l(I_t,y_t^*) - \mathbb{E}\sum_{t=1}^{T} l(\hat{I}_t,y_t^*) =& \sum_{t=1}^{T}\int F_t(\boldsymbol{n})(f(\boldsymbol{n})-f(\boldsymbol{n}-\boldsymbol{l}_t))\mathrm{d}\boldsymbol{n}\\
\leq&\frac{T|E|}{\epsilon},
\end{aligned}$$

which concludes the proof. □

2.1.2 Online MDP Expert Algorithm

As a popular and important way to solve online decision making problems, expert algorithms have been shown perform efficiently in an adversarial MDP scenario. A major problem of using an expert algorithm is that there is no state structure involved in the standard decision making problem. The agent (decision maker) only need to choose an action at each time step. Even-Dar et al. (2003, 2009) proposed the *Markov decision process expert* (MDP-E) algorithm, which associates each state with an expert algorithm. The idea is to train $|S|$ expert algorithms for $|S|$ states. When the agent reach state $\boldsymbol{s}_t$ at time step t, the corresponding expert algorithm $B_{\boldsymbol{s}_t}$ will choose the policy. After the reward function $r_t(\boldsymbol{s}, \boldsymbol{a})$ is observed, $B_{\boldsymbol{s}_t}$ will be adjusted by an appropriate feedback. Even-Dar et al. (2003, 2009) showed that the state-action value function $Q_{r_t}^{\pi_t}$ is a perfect choice, since $Q_{r_t}^{\pi_t}$ provides the "global" information for the decision maker. We can observe the state-action value function is a perfect evaluation of the policy by the following equation

$$\rho_{r_t}(\pi) - \rho_{r_t}(\pi_t) = \mathbb{E}_{\boldsymbol{s}\sim d_\pi}\left[Q_{r_t}^{\pi_t}(\boldsymbol{s}, \pi) - Q_{r_t}^{\pi_t}(\boldsymbol{s}, \pi_t)\right], \tag{2.2}$$

where $\mathbb{E}_{\boldsymbol{s}\sim d_\pi}[\cdot]$ denotes the expectation taken over the stationary state distribution $d_\pi(\boldsymbol{s}) = \lim_{k\to\infty} Pr(\boldsymbol{s}_k = \boldsymbol{s}|\pi)$ generated by the policy π. The notation is slightly abused by writing $Q_r^\pi(\boldsymbol{s}, \pi') = \mathbb{E}_{\boldsymbol{a}\sim\pi'}[Q_r^\pi(\boldsymbol{s}, \boldsymbol{a})]$. Before introducing the formal algorithm, we firstly present an essential assumption for expert algorithm (Even-Dar et al., 2003, 2009).

Assumption 1. *For two arbitrary state distributions $d(\boldsymbol{s})$ and $d'(\boldsymbol{s})$, for all policy π, there exist a positive constant τ such that*

$$\|d(\boldsymbol{s})P^\pi - d'(\boldsymbol{s})P^\pi\|_1 \le e^{-1/\tau}\|d(\boldsymbol{s}) - d'(\boldsymbol{s})\|_1, \tag{2.3}$$

where τ is assumed that $\tau \ge 1$, P^π denotes the $|S| \times |S|$ matrix whose $\boldsymbol{s}\boldsymbol{s}'$th element is $p^\pi(\boldsymbol{s}'|\boldsymbol{s}) = \sum_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s})p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$.

τ is called mixing time which indicates the convergence rate of the MDP converges to its steady status. Then, let us assume that the performance of expert algorithms satisfies the following assumption.

Assumption 2. *An optimized expert algorithm is an algorithm that selects a distribution q_t over action space for any reward function sequence $r_1, \ldots, r_T$, which satisfies*

$$\sum_{t=1}^T \mathbb{E}_{\boldsymbol{a}\sim q_t}\left[r_t(\boldsymbol{a})\right] \le \sup_{\boldsymbol{a}\in A} \sum_{t=1}^T r_t(\boldsymbol{a}) - \sqrt{T \log |A|},$$

where the action distributions q_t do not change quickly:

$$\|q_t - q_{t+1}\|_1 \le \sqrt{\frac{\log |A|}{t}}.$$

A number of expert algorithms satisfy the above assumption, such as the RWM algorithm. Even-Dar et al. (2009) shows that there exists a parameter β such that the RWM is an optimal expert algorithm. Next, we summarize the MDP-E algorithm as shown in Figure 2.3. By Assumption 2, we can obtain the following result by setting the expert algorithm as a black box.

Theorem 2.4. *For any reward function sequence $r_1, \dots, r_T$, the MDP-E algorithm satisfies*

$$L_{\mathrm{MDP-E}}(T) \leq 4\tau^2\sqrt{T \log |A|} + \sqrt{3T\tau \log |A|} + 4\tau.$$

Before showing the proof of Theorem 2.4, we introduce following lemmas:

Lemma 2.5. *For two arbitrary policies π and π', and any arbitrary state distribution d, there is*

$$\|dP^{\pi} - dP^{\pi'}\|_1 \leq \|\pi(\cdot|\boldsymbol{s}) - \pi(\cdot|\boldsymbol{s})\|_1$$

Proof. For all $\boldsymbol{s} \in S$, the transition probabilities induced by π and π' satisfy

$$\begin{aligned}
&\sum_{\boldsymbol{s}' \in S} |p^{\pi}(\boldsymbol{s}'|\boldsymbol{s}) - p^{\pi'}(\boldsymbol{s}'|\boldsymbol{s})| \\
&= \sum_{\boldsymbol{s}' \in S} \sum_{\boldsymbol{a} \in A} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) |\pi(\boldsymbol{a}|\boldsymbol{s}) - \pi'(\boldsymbol{a}|\boldsymbol{s})| \\
&\leq \|\pi(\cdot|\boldsymbol{s}) - \pi'(\cdot|\boldsymbol{s})\|_1.
\end{aligned}$$

Taking expectation over $d(\boldsymbol{s}), \forall \boldsymbol{s} \in S$, we obtain

$$\sum_{\boldsymbol{s} \in S} d(\boldsymbol{s}) \sum_{\boldsymbol{s}' \in S} |p^{\pi}(\boldsymbol{s}'|\boldsymbol{s}) - p^{\pi'}(\boldsymbol{s}'|\boldsymbol{s})| \leq \|\pi(\cdot|\boldsymbol{s}) - \pi'(\cdot|\boldsymbol{s})\|_1$$

□

Lemma 2.6. *For any arbitrary policy π, there is*

$$\|d_{\pi,t} - d_{\pi}\|_1 \leq 2e^{-t/\tau},$$

where d_{π} is the stationary distribution of policy π, $d_{\pi,t}$ is the state distribution at time t following the initial distribution d_1, i.e.

$$d_{\pi,t} = d_1(P^{\pi})^t.$$

```
Put an experts $B_s$ algorithm in every state.
for $t = 1, \ldots, \infty$ do
    Observe current state $\boldsymbol{s}_t$.
    Set the policy $\pi(\boldsymbol{a}|\boldsymbol{s}) = a_t(\boldsymbol{s}_t)$, where $a_t(\boldsymbol{s}_t)$ is the action distribution given by $B_{\boldsymbol{s}_t}$.
    Take action $\boldsymbol{a}_t$ according to $\pi(\boldsymbol{a}|\boldsymbol{s})$.
    Observe reward function $r_t$ and move to $\boldsymbol{s}_{t+1}$.
    Feed $B_{\boldsymbol{s}_t}$ with $Q_{\pi_t, r_t}(\boldsymbol{s}, \cdot)$.
end for
```

Figure 2.3: MDP Expert (MDP-E) Algorithm

Proof. By recurring Eq.(2.3), there is

$$\begin{aligned}\|d_{\pi,t} - d_\pi\|_1 =& e^{-1/\tau}\|d_{\pi,t-1}P^\pi - d_\pi P^\pi\|_1\\ \leq& e^{-1/\tau}\|d_{\pi,t-1} - d_\pi\|_1\\ \leq& e^{-t/\tau}\|d_1 - d_\pi\|_1\\ \leq& 2e^{-t/\tau},\end{aligned}$$

which concludes the proof. □

Lemma 2.7. *After T time steps, the cumulative rewards achieved by the MDP-E algorithm satisfies*

$$\|\sum_{t=1}^{T} \rho_{r_t}(\pi_t) - R_{\mathrm{MDP-E}}(T)\| \leq 4\tau^2\sqrt{\log|A|T} + 2\tau.$$

Proof. By the definition of $R_{\mathrm{MDP-E}}(T)$, there is

$$\begin{aligned}&\sum_{t=1}^{T} \rho_{r_t}(\pi_t) - R_{\mathrm{MDP-E}}(T)\\ &= \sum_{t=1}^{T}\left(\sum_{\boldsymbol{s}\in S}\sum_{\boldsymbol{a}\in A} d_{\pi_t}(\boldsymbol{s})\pi_t(\boldsymbol{a}|\boldsymbol{s})r_t(\boldsymbol{s},\boldsymbol{a}) - \sum_{\boldsymbol{s}\in S}\sum_{\boldsymbol{a}\in A} d_{\mathcal{A},t}(\boldsymbol{s})\pi_t(\boldsymbol{a}|\boldsymbol{s})r_t(\boldsymbol{s},\boldsymbol{a})\right)\\ &\leq \sum_{t=1}^{T}\|d_{\pi_t} - d_{\mathcal{A},t}\|_1,\end{aligned}$$

where the last part can be derived by following results.

$$\begin{aligned}\|d_{\mathcal{A},k} - d_{\pi_t}\|_1 &= \|d_{\mathcal{A},k-1}P^{\pi_k} - d_{\mathcal{A},k-1}P^{\pi_t} + d_{\mathcal{A},k-1}P^{\pi_t} - d_{\pi_t}\|_1 \\ &\leq \|d_{\mathcal{A},k-1}P^{\pi_t} - d_{\pi_t}\|_1 + \|d_{\mathcal{A},k-1}P^{\pi_k} - d_{\mathcal{A},k-1}P^{\pi_t}\|_1 \\ &\leq \|d_{\mathcal{A},k-1}P^{\pi_t} - d_{\pi_t}P^{\pi_t}\|_1 + 2(t-k)\sqrt{\log|A|/t} \\ &\leq e^{-1/\tau}\|d_{\mathcal{A},k-1} - d_{\pi_t}\|_1 + 2(t-k)\sqrt{\log|A|/t}.\end{aligned}$$

Such that

$$\|d_{\mathcal{A},t} - d_{\pi_t}\|_1 \leq 2\tau^2\sqrt{\log|A|/t} + 2e^{-t/\tau},$$

which concludes the proof. □

As we showed earlier, the state-action value function $Q_{r_t}^{\pi_t}(\boldsymbol{s},\boldsymbol{a})$ is used for updating the optimal expert algorithm at each time step. The state-action value function can be treated as the "lost function" which evaluates the current policy. Therefore, we can derive the following bound by Assumption 2 directly:

$$\sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{a}\sim\pi}\left[Q_{r_t}^{\pi_t}(\boldsymbol{s},\boldsymbol{a})\right] - \sum_{t=1}^{T}\mathbb{E}_{\boldsymbol{a}\sim\pi_t}Q_{r_t}^{\pi_t}(\boldsymbol{s},\boldsymbol{a}) \leq \sqrt{3\tau T\log|A|}.$$

Hence, the performance of the MDP-E algorithm satisfies

$$\sum_{t=1}^{T}\rho_{r_t}(\pi) - \sum_{t=1}^{T}\rho_{r_t}(\pi_t) \leq \sqrt{3\tau T\log|A|},$$

which can be obtained by Eq.(2.2). By decomposing the regret as

$$\begin{aligned}L_{\mathrm{MDP-E}}(T) &= R_\pi(T) - R_{\mathrm{MDP-E}}(T) \\ &= \left(R_\pi(T) - \sum_{t=1}^{T}\rho_{r_t}(\pi)\right) + \left(\sum_{t=1}^{T}\rho_{r_t}(\pi) - \sum_{t=1}^{T}\rho_{r_t}(\pi_t)\right) \\ &+ \left(\sum_{t=1}^{T}\rho_{r_t}(\pi_t) - R_{\mathrm{MDP-E}}(T)\right),\end{aligned}$$

we could concludes the proof of Theorem 2.4.

2.1.3 Lazy Follow the Perturbed Leader

In this section, we introduce another expert algorithm based online MDP method called lazy follow the perturbed leader (Lazy-FPL), which is motivated by the FPL algorithm (Cesa-Bianchi and Lugosi, 2006). As same as the FPL algorithm, the Lazy-FPL follows the best policy in the past with a vanishing random perturbation. For decreasing the computation complexity and achieving a sublinear regret, the time horizon is partitioned into phases denoted by $\tau_1, \tau_2, \ldots, \tau_m, \ldots$, where $|\tau_m|$ denotes the length of the mth phase. The lengths of phases control the frequency of switching policies, where the policy should not be switched too often for making the decision stable. On the other hand, the policy should be switched fast enough to adapt the changes of the reward functions. The formal algorithm is presented in Figure 2.4. The regret of the Lazy-FPL satisfies the following theorem.

Theorem 2.8. *After T time steps, the regret of the Lazy-FPL algorithm satisfies*

$$L_{\mathrm{Lazy-FPL}}(T) \leq \frac{4}{3}(2\tau + 2|A| + 4\tau + 1 + 2(|S|+3)|A|^2\tau \log T)T^{-1/4+\epsilon}.$$

Before presenting the proof, let us introduce the following essential lemmas.

Lemma 2.9. *The consecutive polices π_m and π_{m+1} for all $m = 1, 2, \ldots$ satisfy*

$$\|\pi_{m+1}(\cdot|\boldsymbol{s}) - \pi_m(\cdot|\boldsymbol{s})\|_1 \leq (|S|+3)|A|^2 \left(\zeta_{m+1} \frac{|\tau_{m+1}|}{|\tau_{0:m+1}| + \frac{\zeta_{m+1}-\zeta_m}{\zeta_{m+1}}} \right),$$

and

$$\|d_{\pi_{m+1}}\pi_{m+1} - d_{\pi_m}\pi_m\|_1 \leq (|S|+3)|A|^2 \left(\zeta_{m+1} \frac{|\tau_{m+1}|}{|\tau_{0:m+1}| + \frac{\zeta_{m+1}-\zeta_m}{\zeta_{m+1}}} \right) g + 4e^{1-g/\tau},$$

where g is a positive integer.

Proof. Firstly, the solutions of two consecutive linear programming satisfy (Renegar, 1994)

$$|\lambda_{m+1} - \lambda_m| \leq \|\hat{r}_{\tau_{0:m+1}} - \hat{r}_{\tau_{0:m}}\|_1 \leq \frac{|\tau_{m+1}|}{|\tau_{0:m+1}|}, \tag{2.4}$$

$$\|h_{m+1} - h_m\|_\infty \leq 2(|S|+1)\frac{|\tau_{m+1}|}{|\tau_{0:m+1}|}. \tag{2.5}$$

```
Initialize the policy by an arbitrary stationary policy.
for $m = 1, \ldots, \infty$ do
    Solve the linear program:
        $\min_{\lambda \in \mathbb{R}, h \in \mathbb{R}^{|S|}} \quad \lambda$
            $s.t.\ \lambda + h(\boldsymbol{s}) \geq \hat{r}_{\tau_{0:m-1}}(\boldsymbol{s}, \boldsymbol{a}) + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) h(\boldsymbol{s}'), \forall \boldsymbol{s} \in S, \boldsymbol{a} \in A,$
            $h(\boldsymbol{s}^+) = 0$ for some fixed $\boldsymbol{s}^+ \in S$,
    where $\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{|\tau_{0:m-1}|} \sum_{i=1}^{m-1} \sum_{t \in \tau_i} r_t(\boldsymbol{s}, \boldsymbol{a})$ and $|\tau_{0:m-1}| = \sum_{i=1}^{m-1} |\tau_i|$.
    while $t \in \tau_m$ do
        Choose the action
            $\boldsymbol{a}_t = \mathrm{argmax}_{\boldsymbol{a} \in A} \left\{ \hat{r}_{0:m-1}(\boldsymbol{s}, \boldsymbol{a}) + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) h_m(\boldsymbol{s}') + \boldsymbol{n}_m(\boldsymbol{a}) \right\},$
        where $\boldsymbol{n}_m(\boldsymbol{a})$ is the random variable whose probability density function is defined as
            $f_{\boldsymbol{n}_m}(\mathrm{a}) = \begin{cases} \frac{\zeta_m}{2}, & \text{if a} \in [-1/\zeta_m, 1/\zeta_m], \\ 0, & \text{otherwise.} \end{cases}$
    end while
end for
```

Figure 2.4: Lazy Follow the Perturbed Leader (Lazy-FPL) Algorithm

Denote the cumulative distribution functions of two consecutive random variables as $\boldsymbol{n}_{m+1}$ and $\boldsymbol{n}_m$, which satisfy for all $\mathrm{a}, \mathrm{a}' \in \mathbb{R}$

$$|F_{\boldsymbol{n}_{m+1}}(\mathrm{a}) - F_{\boldsymbol{n}_m}(\mathrm{a})| \leq \frac{\zeta_{m+1} - \zeta_m}{2\zeta_{m+1}}, \tag{2.6}$$

and

$$|F_{\boldsymbol{n}_{m+1}}(\mathrm{a}) - F_{\boldsymbol{n}_{m+1}}(\mathrm{a}')| \leq \frac{\zeta_{m+1}}{2}|\mathrm{a} - \mathrm{a}'|. \tag{2.7}$$

The policy $\pi_{m+1}(\boldsymbol{a}|\boldsymbol{s})$ for all $m = 0, \ldots, M$ can be rewritten as

$$\begin{aligned}\pi_{m+1}(\boldsymbol{a}|\boldsymbol{s}) =& Pr\left(\hat{r}_{\tau_{0:m+1}}(\boldsymbol{s}, \boldsymbol{a}) + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) h_m(\boldsymbol{s}') + \boldsymbol{n}_{m+1}(\boldsymbol{a})\right.\\ &\left. > \hat{r}_{\tau_{0:m+1}}(\boldsymbol{s}, \boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}') h_m(\boldsymbol{s}') + \boldsymbol{n}_{m+1}(\boldsymbol{a}'), \forall \boldsymbol{a}' \in A\right)\\ &= \prod_{\boldsymbol{a}' \in A} Pr\left(\boldsymbol{n}_{m+1}(\boldsymbol{a}) - \boldsymbol{n}_{m+1}(\boldsymbol{a}') > \hat{r}_{\tau_{0:m+1}}(\boldsymbol{s}, \boldsymbol{a}') - \hat{r}_{\tau_{0:m+1}}(\boldsymbol{s}, \boldsymbol{a})\right.\\ &\left. + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}') h_m(\boldsymbol{s}') - \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) h_m(\boldsymbol{s}')\right).\end{aligned}$$

Combining Equations (2.4), (2.5), (2.6) and (2.7), we obtained the claimed result of Lemma 2.9. □

Lemma 2.10. *For all $m = 1, 2, \ldots$, the policy π_m satisfies*

$$\rho_{\hat{r}_{\tau_{0:m-1}}}(\pi^*_m) - \rho_{\hat{r}_{\tau_{0:m-1}}}(\pi_m) \leq \frac{2|A|}{\zeta_m^2},$$

where π^*_m is the optimal policy for m phases as

$$\pi^*_m = \mathrm{argmax}_\pi\, \rho_{\hat{r}_{\tau_{0:m-1}}}(\pi).$$

Proof. Similarly to the proof of Lemma 2.10, for all $\boldsymbol{s} \in S$ there is

$$\begin{aligned}
&\pi_m(\boldsymbol{a}|\boldsymbol{s})\\
&\quad =Pr\Bigg(\boldsymbol{n}_m(\boldsymbol{a}) + \hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})h_m(\boldsymbol{s}')\\
&\qquad > \boldsymbol{n}_m(\boldsymbol{a}^*) + \hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}^*) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}^*)h_m(\boldsymbol{s}')\Bigg)\\
&\quad \leq \begin{cases} \begin{matrix}\frac{\zeta_m}{4}(\frac{2}{\zeta_m} - (\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}^*)\\ + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}^*)h_m(\boldsymbol{s}')\\ -\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a})\\ + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})h_m(\boldsymbol{s}')))^2,\end{matrix} & if \quad \begin{matrix}\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}^*)\\ + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}^*)h_m(\boldsymbol{s}')\\ -\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a})\\ + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})h_m(\boldsymbol{s}')) \leq \frac{2}{\zeta_m}\end{matrix}\\ 0, & otherwise. \end{cases}
\end{aligned}$$

By rewriting the policy as above expression, we can obtain

$$\begin{aligned}
\rho_{\hat{r}_{\tau_{0:m-1}}}(\pi^*_m) - \rho_{\hat{r}_{\tau_{0:m-1}}}(\pi_m) \leq& \max_{\boldsymbol{s}\in S} \sum_{\boldsymbol{a}\neq\boldsymbol{a}^*} (\hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}^*) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}^*)h_m(\boldsymbol{s}')\\
&- \hat{r}_{\tau_{0:m-1}}(\boldsymbol{s},\boldsymbol{a}) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})h_m(\boldsymbol{s}'))\pi_m(\boldsymbol{a}|\boldsymbol{s})\\
\leq& \frac{2|A|}{\zeta_m^2},
\end{aligned}$$

which concludes the proof. □

Following the result of Lemma 2.9, we obtain the following results by setting $\zeta_m = \sqrt{|\tau_{0:m}|}$ and $g = \tau \log(|\tau_{0:m+1}|)$

$$\|d_{\pi_{m+1}}\pi_{m+1} - d_{\pi_m}\pi_m\|_1 \leq 2(|S|+3)|A|^2\tau\frac{|\tau_{m+1}|\log(|\tau_{0:m+1}|)}{|\tau_{0:m+1}|} + \frac{4}{|\tau_{0:m+1}|}$$

Given the fact $|\tau_m||\tau_{m+1}|\log(|\tau_{0:m+1}|) \leq \log(T)|\tau_{0:m+1}|^{1/2}$, the sum of the above inequality satisfies

$$\sum_{m=0}^{M-1}\sum_{t\in\tau_m}\rho_{r_t}(\pi_{m+1}) - \sum_{m=0}^{M-2}\sum_{t\in\tau_m}\rho_{r_t}(\pi_m) \leq 2(|S|+3)|A|^2\tau\log T + 4. \qquad (2.8)$$

Since π_{m+1} is the perturbed optimal policy over m phases, the effect of the perturbed random variables is bounded as

$$\sum_{m=0}^{M-2}\sum_{t\in\tau_m}\rho_{r_t}(\pi_{M+1}) \leq \sum_{m=0}^{M-2}\sum_{t\in\tau_m}\rho_{r_t}(\pi_{m+1}) + 2(M-2)|A|.$$

The above inequality can be obtained by induction. It is clear that the following statement holds with $M = 2$,

$$\rho_{r_0}(\pi_1) = \rho_{r_0}(\pi_1).$$

Then we assume that for some positive integer M, there is

$$\sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi_{\mathrm{M}+1}) \leq \sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi_{m+1}) + 2\mathrm{M}|A|.$$

Thus, we obtain the following result with $\mathrm{M} + 1$,

$$\begin{aligned}\sum_{m=0}^{\mathrm{M}+1} \sum_{t \in \tau_m} \rho_{r_t}(\pi_{m+1}) &\geq \sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi_{m+1}) \\ &\geq \sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi_{\mathrm{M}+1}) - 2\mathrm{M}|A| \\ &\geq \sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi^*_{\mathrm{M}+1}) - 2\mathrm{M}|A| - 2|A|\frac{|\tau_{0:\mathrm{M}}|}{\zeta^2_{\mathrm{M}+1}} \\ &\geq \sum_{m=0}^{\mathrm{M}} \sum_{t \in \tau_m} \rho_{r_t}(\pi^*_{\mathrm{M}+2}) - 2\mathrm{M} + 1|A|. \end{aligned} \tag{2.9}$$

Therefore, we obtain the claimed result of Theorem 2.8 by combining Equ.(2.8) and Equ.(2.9),

$$\begin{aligned}\sup_{\pi} \sum_{t=1}^{T} \rho_{r_t}(\pi) \leq& \sum_{m=0}^{T} \sum_{t \in \tau_m} \rho_{r_t}(\pi_m) \\ &+ (M-1)(4 + 2(|S|+3)|A|^2 \tau \log(T)) + 2(M-1)|A| + M^{1/3}.\end{aligned}$$

2.2 Online MDPs with Online Linear Optimization

In this section, we introduce another class of online MDP algorithms which use online linear optimization techniques. By observing the objective function of regret minimization, the online MDP problems can be simplified as linear learning problems by using the notion of the stationary occupancy measure. First of all, we review two major online convex optimization algorithms. Then we show the mechanism of solving online MDPs with online linear optimization. Moreover, we provide an analysis for some state of the art algorithms with explicit regret bounds.

2.2.1 Online Convex Optimization

Online convex optimization problems are a special kind of online decision making problems involving a convex compact set and a set of convex cost functions. Formally, the online convex optimization problem is defined as follows:

- for $t = 1$ to ∞
    1. Select a vector $\boldsymbol{x}_t \in X$, where $X \subset \mathbb{R}^d$ is a convex compact vectors set.
    2. Reveal the convex cost function $c_t : X \to [0, 1]$ to the decision maker.
    3. Suffer cost $c_t(\boldsymbol{x}_t)$.

Recall the expert prediction algorithm, the regret is defined with respect to a best expert over the experts set. Similarly, the regret of the online convex optimization algorithm $\mathcal{A}$ with respect to X over T time steps is defined as

$$R_{\mathcal{A}}(T) = \sum_{t=1}^{T} c_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in X} \sum_{t=1}^{T} c_t(\boldsymbol{x}).$$

In the rest of this section, we introduce two algorithms: the *online gradient descent* (OPG) algorithm (Zinkevich, 2003) and the *online mirror descent* (OMD) algorithm (Beck and Teboulle, 2003; Shalev-Shwartz, 2011).

Zinkevich (2003) proposed a gradient descent based algorithm for online convex optimization with greedy projection. The main idea is to apply gradient descent in $\mathbb{R}^d$ real space, and project the vector back to X. At time step t, after the cost function c_t is revealed the prediction vector is updated as

$$\boldsymbol{x}_{t+1} = P(\boldsymbol{x}_t - \eta_t \nabla c_t(\boldsymbol{x}_t)),$$

where η_t is the step size. The greedy projection P is defined as

$$P(\tilde{\boldsymbol{x}}) = \operatorname{argmin}_{\boldsymbol{x} \in X} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2.$$

Before analyzing the regret, we introduce the assumptions required in the analysis:

1. X is compact bounded nonempty set.

2. The cost functions are differentiable. The gradient ∇c_t satisfies

$$\nabla c_t(\boldsymbol{x}) \leq G, \forall \boldsymbol{x} \in X, t = 1, \ldots, \infty$$

Theorem 2.11. *Let $\eta_t = \frac{1}{\sqrt{t}}, \forall t = 1, \ldots, T$, the regret of the OGD algorithm over T time steps is bounded as*

$$L_{\mathrm{OGD}}(T) \leq \frac{|X|^2\sqrt{T}}{2} + (\sqrt{T} - \frac{1}{2})G^2.$$

Proof. The proof is provided in Zinkevich (2003). Since for all $t = 1, \ldots, T$ the cost function c_t is convex, then we can obtain the following result from the definition of convexity:

$$c_t(\boldsymbol{x}) - c_t(\boldsymbol{x}_t) \geq (\nabla c_t(\boldsymbol{x}_t)) \cdot (\boldsymbol{x} - \boldsymbol{x}_t), \forall \boldsymbol{x} \in X.$$

Note that the greedy projection P satisfies $(P(\tilde{\boldsymbol{x}}) - \boldsymbol{x})^2 \leq (\tilde{\boldsymbol{x}} - \boldsymbol{x})^2$. Then we can obtain

$$\begin{aligned}(\boldsymbol{x}_{t+1} - \boldsymbol{x})^2 &\leq (\boldsymbol{x}_t - \eta_t \nabla c_t(\boldsymbol{x}_t) - \boldsymbol{x})^2 \\ &\leq (\boldsymbol{x}_t - \boldsymbol{x})^2 + \eta_t^2 G^2 - 2(\boldsymbol{x}_t - \boldsymbol{x})\eta_t \nabla c_t(\boldsymbol{x}_t).\end{aligned}$$

By rearranging the above result and defining the best vector $\boldsymbol{x}^*$ as

$$\boldsymbol{x}^* = \mathrm{argmin}_{\boldsymbol{x} \in X} \sum_{t=1}^{T} c_t(\boldsymbol{x}),$$

we get

$$(\boldsymbol{x}_t - \boldsymbol{x})\nabla c_t(\boldsymbol{x}_t) \leq \frac{1}{2\eta_t}\left((\boldsymbol{x}_t - \boldsymbol{x})^2 - (\boldsymbol{x}_{t+1} - \boldsymbol{x})^2\right) + \frac{\eta_t}{2}G^2.$$

Then the regret can be bounded as

$$
\begin{aligned}
L_{\mathrm{OGD}}(T) =& \sum_{t=1}^{T} (c_t(\boldsymbol{x}_t) - c_t(\boldsymbol{x}^*)) \\
\leq& \sum_{t=1}^{T} ((\boldsymbol{x}_t - \boldsymbol{x}^*) \cdot \nabla c_t(\boldsymbol{x}_t)) \\
\leq& \sum_{t=1}^{T} \frac{1}{2\eta_t} \left((\boldsymbol{x}_t - \boldsymbol{x}^*)^2 - (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^2\right) + \frac{\eta_t}{2} G^2 \\
\leq& \frac{1}{2\eta_1}(\boldsymbol{x}_1 - \boldsymbol{x}^*)^2 - \frac{1}{2\eta_T}(\boldsymbol{x}_{T+1} - \boldsymbol{x}^*)^2 + \frac{1}{2}\sum_{t=1}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)(\boldsymbol{x}_t - \boldsymbol{x}^*)^2 \\
&+ \frac{G^2}{2}\sum_{t=1}^{T} \eta_t \\
\leq& |X|^2 \frac{1}{2\eta_T} + \frac{G^2}{2}\sum_{t=1}^{T} \eta_t
\end{aligned}
$$

Judiciously setting the step size as $\eta_t = \frac{1}{\sqrt{t}}$, we can conclude the proof. □

Now we move to a sharper regret bound of the OGD algorithm with a stronger assumption (Hazan et al., 2007):

Theorem 2.12. *If the cost functions $c_t, t = 1, \ldots, T$ are H-strong convex, the second derivatives of the cost functions satisfy*

$$\nabla^2 c_t(\boldsymbol{x}) \succeq H\boldsymbol{I}_d,$$

where $\boldsymbol{I}_d$ is the d-dimensional identity matrix. Then the regret bound of OGD algorithm over T trails satisfies

$$L_{\mathrm{OGD}}(T) \leq \frac{G^2}{2H}(1 + \log T),$$

where the step size is $\eta_t = \frac{1}{Ht}$.

Proof. The basic idea of this proof (Hazan et al., 2007) follows the same line with Zinkevich (2003)'s analysis. Firstly, we expand the cost function by Taylor's theorem as

$$\begin{aligned} c_t(\boldsymbol{x}^*) =& c_t(\boldsymbol{x}_t) + \nabla c_t(\boldsymbol{x}_t) \cdot (\boldsymbol{x}^* - \boldsymbol{x}_t) + \frac{1}{2}(\boldsymbol{x}^* - \boldsymbol{x}_t)^\top \nabla^2 c_t(\zeta_t)(\boldsymbol{x}^* - \boldsymbol{x}_t) \\ \geq& c_t(\boldsymbol{x}_t) + \nabla c_t(\boldsymbol{x}_t) \cdot (\boldsymbol{x}^* - \boldsymbol{x}_t) + \frac{H}{2}(\boldsymbol{x}^* - \boldsymbol{x}_t)^\top (\boldsymbol{x}^* - \boldsymbol{x}_t), \end{aligned}$$

where ζ_t is a vector between $\boldsymbol{x}^*$ and $\boldsymbol{x}_t$, such that the above expansion holds. By rearranging this result, we obtain

$$c_t(\boldsymbol{x}_t) - c_t(\boldsymbol{x}^*) \leq \nabla c_t(\boldsymbol{x}_t) \cdot (\boldsymbol{x}^* - \boldsymbol{x}_t) - \frac{H}{2}(\boldsymbol{x}^* - \boldsymbol{x}_t)^\top (\boldsymbol{x}^* - \boldsymbol{x}_t).$$

Thus,

$$\begin{aligned} L_{\text{OGD}}(T) =& \sum_{t=1}^{T} \left(c_t(\boldsymbol{x}_t) - c_t(\boldsymbol{x}^*)\right) \\ \leq& \sum_{t=1}^{T} \frac{1}{2\eta_t}\left((\boldsymbol{x}_t - \boldsymbol{x}^*)^2 - (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^2\right) + \frac{\eta_t}{2}G^2 - \frac{H}{2}(\boldsymbol{x}_t - \boldsymbol{x}^*)^2 \\ \leq& \frac{G^2}{2}\sum_{t=1}^{T}\eta_t, \end{aligned}$$

which concludes the proof. □

Considering the generalization of the OGD algorithm, the online mirror descent (OMD) algorithm is proposed for solving online convex optimization problems. More precisely, the online mirror descent solves online convex optimization as a regularized minimization which can be expressed as

$$\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in X}\left(\eta \sum_{i=1}^{t} c_i(\boldsymbol{x}) + R(\boldsymbol{x})\right).$$

Here, $R(\boldsymbol{x})$ is the regularization which keeps the predictions stable. Note that the trade off between minimizing $c_t(\boldsymbol{x})$ and keeping $\boldsymbol{x}_{t+1}$ close to $\boldsymbol{x}_t$. By this expression, we can fit the OGD algorithm into the OMD framework with $R(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|^2}{2}$. However, we will consider another distance measure “Bregman Divergence” instead of the l_2 norm. The Bregman divergence is defined with respect to a Legendre function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$D_f(\boldsymbol{a}, \boldsymbol{b}) = f(\boldsymbol{a}) - f(\boldsymbol{b}) - \nabla f(\boldsymbol{b}) \cdot (\boldsymbol{a} - \boldsymbol{b}), \forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d.$$

At each time step, a regularized minimization is solved and the prediction is "mirrored" to the feasible set X via the Bregman projection. The update rule is given as follows:

- Initialize $\tilde{\boldsymbol{x}}_1 = 0$.
- for $t = 1$ to ∞
  1. Update the prediction vector as
  $$\boldsymbol{x}_t = \mathrm{P}_R(\tilde{\boldsymbol{x}}_t),$$
  where $\mathrm{P}_R(\boldsymbol{x}') = \mathrm{argmin}_{\boldsymbol{x} \in X} D_R(\boldsymbol{x}, \boldsymbol{x}')$, $D_R(\cdot, \cdot)$ is the Bregman divergence defined with respect to the regularization function.
  2. Update $\tilde{\boldsymbol{x}}$ as
  $$\tilde{\boldsymbol{x}}_{t+1} = (\nabla R)^{-1}(\nabla R(\tilde{\boldsymbol{x}}_t) - \eta \nabla c_t(\tilde{\boldsymbol{x}}_t)).$$

The regret of the OMD algorithm can be summarized as following theorem:

Theorem 2.13. *After T time steps, the regret of the OMD algorithm is bounded as*

$$L_{\mathrm{OMD}}(T) \leq \frac{D_R(\boldsymbol{x}, \boldsymbol{x}_1)}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} D_R(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}).$$

Proof. The proof we present follows Cesa-Bianchi and Lugosi (2006). Firstly, let us show that the update rule of the OMD algorithm leads us to the solution of the following unconstrained minimization:

$$\tilde{\boldsymbol{x}}_{t+1} = \mathrm{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \left(\eta \sum_{i=1}^{t} \nabla c_i(\tilde{\boldsymbol{x}}_i) \cdot \boldsymbol{x} + R(\boldsymbol{x}) \right).$$

By taking the derivative and setting it to zero, we have

$$\nabla R(\tilde{\boldsymbol{x}}_{t+1}) = -\eta \sum_{i=1}^{t} \nabla c_i(\tilde{\boldsymbol{x}}_i),$$
$$\nabla R(\tilde{\boldsymbol{x}}_t) = -\eta \sum_{i=1}^{t-1} \nabla c_i(\tilde{\boldsymbol{x}}_i).$$

Thus we obtain $\nabla R(\tilde{\boldsymbol{x}}_{t+1}) = \nabla R(\tilde{\boldsymbol{x}}_t) - \eta \nabla c_t(\tilde{\boldsymbol{x}}_t)$, which is equivalent to the update rule of the OMD algorithm. Let us define the potential functions as

$$\phi_0 = R, \phi_{t+1} = \phi_t + \eta \nabla c_{t+1}(\tilde{\boldsymbol{x}}_{t+1}).$$

It is clear that

$$\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \phi_t(\boldsymbol{x}),$$

so that

$$\nabla \phi_t(\tilde{\boldsymbol{x}}_{t+1}) = \nabla \phi_{t-1}(\tilde{\boldsymbol{x}}_t) = \ldots = \nabla \phi_0(\tilde{\boldsymbol{x}}_1) = 0.$$

Since the cost functions are convex, we have

$$\sum_{t=1}^{T} (c_t(\boldsymbol{x}_t) - c_t(\boldsymbol{x})) \leq \sum_{t=1}^{T} \nabla c_t(\boldsymbol{x}_t) \cdot (\boldsymbol{x}_t - \boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^d.$$

Thus, we obtain the following results

$$\begin{aligned}
\sum_{t=1}^{T} \nabla c_t(\tilde{\boldsymbol{x}}_t) \cdot (\tilde{\boldsymbol{x}}_t - \boldsymbol{x}) =& \sum_{t=1}^{T} \nabla c_t(\tilde{\boldsymbol{x}}_t) \cdot \tilde{\boldsymbol{x}}_t - \phi_T(\boldsymbol{x}) + R(\boldsymbol{x}) \\
\leq& \sum_{t=1}^{T} \nabla c_t(\tilde{\boldsymbol{x}}_t) \cdot \tilde{\boldsymbol{x}}_t - \phi_T(\tilde{\boldsymbol{x}}_{T+1}) + R(\boldsymbol{x}) \\
\leq& \sum_{t=1}^{T} (\phi_t(\tilde{\boldsymbol{x}}_t) - \phi_t(\tilde{\boldsymbol{x}}_{t+1})) + R(\boldsymbol{x}) - R(\tilde{\boldsymbol{x}}_1) \\
=& D_R(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{x}}_{t+1}) + D_R(\boldsymbol{x}, \tilde{\boldsymbol{x}}_1).
\end{aligned}$$

By the Bregman projection, the constrained minimization is equivalent to the unconstrained minimization as

$$\boldsymbol{x}_{t+1} = \mathrm{P}_R(\tilde{\boldsymbol{x}}_{t+1}),$$

where $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in X} \phi_t(\boldsymbol{x})$, $\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x}_{\mathbb{R}^d}} \phi_t(\boldsymbol{x})$, such that we obtain the claimed result. □

2.2.2 MD^2 for Online MDP

As we have shown in Section 2.1, the regret of any online MDP algorithm $\mathcal{A}$ can be decomposed as follows,

$$L_{\mathcal{A}}(T) \leq \sum_{t=1}^{T} \rho_{r_t}(\pi) - \sum_{t=1}^{T} \rho_{r_t}(\pi_t) + T(\tau+1)\delta + 4\tau + 4, \tag{2.10}$$

where $\mathbb{E}[\|d_{\pi_t} - d_{\pi_{t-1}}\|_1], \forall t = 2, \ldots, T$, $\pi_t, t = 1, \ldots, T$ are given by $\mathcal{A}$. Since Assumption 1 gives the convergence rate of the state distribution to the stationary distribution, the regret minimization could be done by finding a slowly changing stationary state distribution sequence. By the definition of $\rho_{r_t}(\pi)$, the first part of the above inequality is the regret of a linear optimization problem as

$$\sum_{t=1}^{T} \sum_{\boldsymbol{s}\in S} \sum_{\boldsymbol{a}\in A} r_t(\boldsymbol{s},\boldsymbol{a}) d_{\pi}(\boldsymbol{s},\boldsymbol{a}) - \sum_{t=1}^{T} \sum_{\boldsymbol{s}\in S} \sum_{\boldsymbol{a}\in A} r_t(\boldsymbol{s},\boldsymbol{a}) d_{\pi_t}(\boldsymbol{s},\boldsymbol{a}).$$

Therefore, the online MDP problem is equivalent to an online linear optimization problem consisting of following components:

- A convex feasible set $\mathrm{X} \subset [0,1]^{|S|\times|A|}$, which satisfies

$$\mathrm{K} = \Big\{ \mu \in [0,1]^{|S|\times|A|} : \sum_{\boldsymbol{s}\in S, \boldsymbol{a}\in A} \mu(\boldsymbol{s},\boldsymbol{a}) = 1, \\ \sum_{\boldsymbol{a}'\in A} \mu(\boldsymbol{s}',\boldsymbol{a}') = \sum_{\boldsymbol{s}\in S, \boldsymbol{a}\in A} \mu(\boldsymbol{s},\boldsymbol{a}) p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}), \forall \boldsymbol{s}' \in S \Big\}.$$

- An infinite sequence of linear functions $\{\langle r_1, \mu\rangle, \langle r_2, \mu\rangle, \ldots\}$, where $\langle r_t, \mu\rangle = \sum_{\boldsymbol{s}\in S, \boldsymbol{a}\in A} r_t(\boldsymbol{s},\boldsymbol{a})\mu(\boldsymbol{s},\boldsymbol{a})$ for $t = 1, 2, \ldots$.

The feasible set K is the set of the stationary occupancy measures induced by all the Markovian policies. At each time step t, an online linear optimization algorithm selects a point $\mu_t \in \mathrm{K}$. After T time steps, the sequence $\mu_t, t = 1, 2, \ldots, T$ satisfies

$$\sum_{t=1}^{T} \langle r_t, \mu\rangle - \sum_{t=1}^{T} \langle r_t, \mu_t\rangle = o(T).$$

By Eq.(2.10), we observe that any online linear optimization algorithm also achieves a sublinear regret for online MDP problems as long as the decision vector does not change frequently.

As we have shown in the previous section, many online convex optimization algorithms could be proved to perform well for the online MDP problems, e.g. the online gradient descent method and the online mirror descent method. Next we present a specific algorithm called MD^2 for online MDPs based on the online mirror descent.

MD^2 is an efficient implementation of the mirror descent algorithm with approximate projections. Dick et al. (2014) used the second order regularizer $R(\mu) = \frac{1}{2}\|\mu\|_2^2$, which reduces the Bregman divergence and the Bregman projection to the Euclidean distance and the Euclidean projection, respectively. The benefit of this regularizer is that the approximate projection can be obtained as the solution of a quadratic programming problem. Before showing the regret result, we introduce an additional assumption.

Assumption 3. *The stationary occupancy measures induced by all the Markovian policies and the transition probabilities are lower bounded by $\beta > 0$. Such that the feasible set K satisfies*

$$\mathrm{K} = \Big\{\mu \in [\beta, 1]^{|S|\times|A|} : \sum_{\boldsymbol{s}\in S, \boldsymbol{a}\in A} \mu(\boldsymbol{s}, \boldsymbol{a}) = 1, \\ \sum_{\boldsymbol{a}'\in A} \mu(\boldsymbol{s}', \boldsymbol{a}') = \sum_{\boldsymbol{s}\in S, \boldsymbol{a}\in A} \mu(\boldsymbol{s}, \boldsymbol{a}) p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}), \forall \boldsymbol{s}' \in S \Big\}.$$

- Initialize $\mu_1 \in \mathrm{K}$ by arbitrary point, where $\mathrm{K} \subset [\beta, 1]^{|S|\times|A|}$.
- For $t = 1, \ldots, \infty$

  1. Update the $\tilde{\mu}_{t+1}$ by

     $$\tilde{\mu}_{t+1} = \mathrm{argmin}_{\mu\in\mathbb{R}^{|S|\times|S|}} \left(\eta\langle l_t, \mu\rangle + D_R(\mu, \mu_t)\right),$$

     which can be expressed in closed form efficiently.

  2. Project $\tilde{\mu}_{t+1}$ back into the feasible set by the projection

     $$\mu_{t+1} = \mathrm{P}_R(\tilde{\mu}_{t+1}),$$

     which can be approximated by the c-approximate projections satisfies

     $$\|\mu_{t+1} - \mathrm{P}_R(\tilde{\mu}_{t+1})\|_1 \leq c.$$

The performance of the MD^2 algorithm on online MDP problems is analyzed in the following theorem.

Theorem 2.14. *For any* $\mu_1 \in \mathrm{K}$*, the regret of the* MD^2 *algorithm after* T *time steps satisfies*

$$L_{\mathrm{MD}^2}(T) \leq 2\sqrt{(2\tau+3)TD_R(\mu,\mu_1)} + 2\sqrt{T} + 4\tau + 4,$$

where $\eta = \sqrt{\frac{D_R(\mu,\mu_1)}{T(2\tau+3)}}$ *and* $c = \frac{\beta\eta}{T}$.

Proof. The proof of Theorem 2.14 follows Equ.(2.10) with the following lemma.

Lemma 2.15. *After* T *time steps, the* MD^2 *algorithm gives the prediction sequence* $\mu_1, \mu_2, \ldots, \mu_T$ *for the online linear optimization problem with reward functions* $r_1, \ldots, r_T$*. Then for any* $\mu \in \mathrm{K}$*, there is*

$$\sum_{t=1}^{T} \langle r_t, \mu - \mu_t \rangle \leq \sum_{t=1}^{T} \langle r_t, \mu_t - \tilde{\mu}_{t+1} \rangle + \frac{D_R(\mu,\mu_1)}{\eta} + \frac{cT}{\beta\eta},$$

where $\tilde{\mu}_{t+1} = \mu_t e^{-\eta r_t}$.

The proof of Lemma 2.15 can be directly obtained by Theorem 2.13. □

Chapter 3

Online Policy Gradient for Continuous States and Actions Online MDPs

We consider the learning problem under an online Markov decision process (MDP), which is aimed at learning the time-dependent decision-making policy of an agent that minimizes the regret — the difference from the best fixed policy. The difficulty of online MDP learning is that the reward function changes over time. In this chapter, we show that a simple online policy gradient algorithm performs asymptotically equal to the best fixed policy by parameterizing the policy space. Furthermore, it achieves regret $O(\sqrt{T})$ for T steps under a certain concavity assumption and $O(\log T)$ under a strong concavity assumption. To the best of our knowledge, this is the first work to give an online MDP algorithm that can handle continuous state, action, and parameter spaces with guarantee. We also illustrate the behavior of the proposed online policy gradient method through experiments.

3.1 Introduction

As shown in Section 2.1, the MDP expert algorithm (MDP-E), which chooses the current best action at each state, was shown to achieve regret $O(\sqrt{T \log |A|})$ (Even-Dar et al., 2003, 2009), where $|A|$ denotes the cardinality of the action space. Although this bound does not explicitly depend on the cardinality of the state space, the algorithm itself needs an expert algorithm for each state, and thus large state space may not be handled in practice. The lazy follow-the-perturbed-leader (lazy-FPL) divides the time steps into short periods and policies are updated only at the end of each period using the average reward function (Yu et al., 2009). This lazy-FPL algorithm was shown to have regret $O(T^{3/4+\epsilon}\log T(|S|+|A|)|A|^2)$ for $\epsilon \in (0, 1/3)$. The online MDP problem is formulated as an online linear optimization problem in Dick et al. (2014). By introducing the stationary occupation measures, the *mirror descent with approximate projections* was shown to have regret $O(\sqrt{T})$. However, the algorithm assumes that both the state and action spaces are finite. Furthermore it is not straightforward to extend their theoretical results into continuous problems without additional assumptions. Yu et al. (2009), Abbasi-Yadkori et al. (2013), and Neu et al. (2012) considered even more challenging online MDP problems under unknown or changing transition dynamics.

In many real problems, full information of the reward function may be hard to acquire, but only the value of the reward function for the current state and action is available. Such a setup, called the *bandit feedback* scenario, has attracted a great deal of attention recently. An extension of the lazy-FPL method to the bandit feedback scenario, called the *exploratory-FPL* algorithm (Yu et al., 2009), was shown to have regret $o(T)$. Neu et al. (2010b) proposed a method based on MDP-E that uses an unbiased estimator of the reward function, and showed that its regret is $O(T^{2/3}(\ln T)^{1/3}\ln|A|)$. Neu et al. (2014) further improved the regret bound to $O(\sqrt{T\ln T\ln|A|})$. However, this algorithm cannot be used in continuous state and action problems.

In this chapter, we propose an *online policy gradient* (OPG) algorithm that can be implemented in a straightforward manner for problems with continuous state and action spaces. Under the assumption that the expected average reward function is concave, we prove that the regret of our OPG algorithm with respect to a compact and convex parametric policies set is $O(\sqrt{T}(F^2+N))$, which is independent of the cardinality of the state and action spaces, but is dependent on the diameter F and dimension N of the parameter space. Furthermore, regret $O(N^2\log T)$ is also proved under a strong concavity assumption on the expected average reward function. We also extend the proposed algorithm to a bandit feedback scenario, and theoretically prove that the regret bound of the proposed algorithm is $O(\sqrt{T})$ with the concavity assumption. We numerically illustrate the superior behavior of the proposed OPG algorithm in continuous problems over MDP-E with different discretization schemes.

3.2 Online Policy Gradient (OPG) algorithm

In this section, we firstly present involved preliminaries of the online MDP problem. Then we introduce the proposed online policy gradient algorithm with full information and bandit feedback.

3.2.1 Preliminaries

An online MDP algorithm produces a stochastic time-dependent policy, which is a conditional probability density of action $\boldsymbol{a}$ to be taken given current state $\boldsymbol{s}$ at each time step. In this chapter, we suppose that the online MDP algorithm $\mathcal{A}$ outputs parameter $\boldsymbol{\theta}_t = [\theta_t^{(1)}, \dots, \theta_t^{(N)}]^\top \in \Theta \subset \mathbb{R}^N$ of stochastic policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)$ at each time step t, where Θ is a convex and compact parameter set. Thus, algorithm $\mathcal{A}$ gives a sequence of policies:

$$\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_1), \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_2), \dots, \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_T).$$

Ideally, the objective is to maximize the expected cumulative reward over T time steps of algorithm $\mathcal{A}$, which can be denoted as

$$R_{\mathcal{A}}(T) = \mathbb{E}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\middle|\mathcal{A}\right]. \tag{3.1}$$

In above definition, $\mathbb{E}[\cdot|\mathcal{A}]$ denotes the expectation over the joint state-action distribution $p_t(\boldsymbol{s}, \boldsymbol{a}|\mathcal{A})$ given the algorithm $\mathcal{A}$ has been followed at each time step. The state-action distribution induced by $\mathcal{A}$ and the transition density at time step t can be expressed as

$$p_t(\boldsymbol{s}, \boldsymbol{a}|\mathcal{A}) = d_{\mathcal{A},t}(\boldsymbol{s}) \cdot \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t),$$

where the state distribution induced by $\mathcal{A}$ at time step t is defined as

$$d_{\mathcal{A},t}(\boldsymbol{s}) = p(\boldsymbol{s}_t = \boldsymbol{s}|\mathcal{A}).$$

As we mentioned earlier, maximizing the objective defined in Eq.(3.1) is not possible, since we cannot observe all T reward functions during the process of online decision making problem. Here, we instead design algorithm $\mathcal{A}$ that minimizes the *regret* against the baseline which is the best parametric offline policy defined by

$$L_{\mathcal{A}}(T) = R_{\boldsymbol{\theta}^*}(T) - R_{\mathcal{A}}(T).$$

In above definition of the regret, we suppose that there exists $\boldsymbol{\theta}^*$ such that policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}^*)$ maximizes the expected cumulative rewards:

$$R_{\boldsymbol{\theta}^*}(T) = \mathbb{E}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\middle|\boldsymbol{\theta}^*\right].$$

The best offline parameter $\boldsymbol{\theta}^*$ is given by

$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}\in\Theta} \mathbb{E}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\middle|\boldsymbol{\theta}\right], \tag{3.2}$$

where $\mathbb{E}[\cdot|\boldsymbol{\theta}]$ denotes the expectation over the state-action distribution given the policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})$ has been followed at each time step.

In this chapter, we assume that all candidate policies are parameterized by the parameter $\boldsymbol{\theta}$, which is different from related works with finite states and actions (Even-Dar et al., 2003, 2009; Neu et al., 2010b; Yu et al., 2009; Dick et al., 2014). For continuous problems, it is a common choice to use a parametric policy (e.g., the Gaussian policy) which was demonstrated to work well (Sutton and Barto, 1998; Peters and Schaal, 2006). For this reason, the best offline policy defined in Eq.(3.2) is a suitable baseline given that the best policy with respect to the class of all Markovian policies is not a suitable baseline for continuous problems. If the regret is bounded by a sub-linear function with respect to T, the algorithm $\mathcal{A}$ is shown to be asymptotically as powerful as the best offline policy.

3.2.2 OPG Algorithm

Differently from the previous works (Even-Dar et al., 2003, 2009; Neu et al., 2010b), we do not use the expert algorithm in our method, because it is not suitable for handling continuous state and action problems. Instead, we consider a gradient-based algorithm which updates the parameter of policy $\boldsymbol{\theta}$ along the gradient direction of the expected average reward function at each time step t.

More specifically, we assume that all the MDPs are ergodic whose state transitions are induced by the transition density $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ and the parametric policy $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \Theta$. Then every policy $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta})$ has a unique stationary state distribution $d_{\boldsymbol{\theta}}(\boldsymbol{s})$:

$$d_{\boldsymbol{\theta}}(\boldsymbol{s}) = \lim_{t \to \infty} p(\boldsymbol{s}_t = \boldsymbol{s}|\boldsymbol{\theta}).$$

Note that the stationary state distribution satisfies

$$d_{\boldsymbol{\theta}}(\boldsymbol{s}') = \int_{\boldsymbol{s} \in S} d_{\boldsymbol{\theta}}(\boldsymbol{s}) \int_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta}) p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) \mathrm{d}\boldsymbol{a} \mathrm{d}\boldsymbol{s}.$$

Let $\rho_{r_t}(\boldsymbol{\theta})$ be the expected average reward function of policy $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta})$ at time step t:

$$\begin{aligned} \rho_{r_t}(\boldsymbol{\theta}) =& \mathbb{E}_{\boldsymbol{s} \sim d_{\boldsymbol{\theta}}(\boldsymbol{s}), \boldsymbol{a} \sim \pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta})} \left[r_t(\boldsymbol{s}, \boldsymbol{a}) \right] \\ =& \int_{\boldsymbol{s} \in S} d_{\boldsymbol{\theta}}(\boldsymbol{s}) \int_{\boldsymbol{a} \in A} r_t(\boldsymbol{s}, \boldsymbol{a}) \pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{a} \mathrm{d}\boldsymbol{s}, \end{aligned} \tag{3.3}$$

where the expectation is taken over the stationary state-action distribution of policy $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta})$.

Then our *online policy gradient (OPG) algorithm* is given as follows:

- Initialize policy parameter $\boldsymbol{\theta}_1$.
- for $t = 1$ to ∞
  1. Observe current state $\boldsymbol{s}_t = \boldsymbol{s}$.
  2. Take action $\boldsymbol{a}_t = \boldsymbol{a}$ according to current policy $\pi(\boldsymbol{a}|\boldsymbol{s}; \boldsymbol{\theta}_t)$.
  3. Observe reward r_t from the environment.
  4. Move to next state $\boldsymbol{s}_{t+1}$.
  5. Update the policy parameter as

$$\boldsymbol{\theta}_{t+1} = P\left(\boldsymbol{\theta}_t + \eta_t \nabla_{\boldsymbol{\theta}} \rho_{r_t}(\boldsymbol{\theta}_t)\right), \tag{3.4}$$

where $P(\boldsymbol{\vartheta}) = \arg\min_{\boldsymbol{\theta}\in\Theta} \|\boldsymbol{\vartheta} - \boldsymbol{\theta}\|$ is the projection function on parameter space, $\|\cdot\|$ denotes the Euclidean norm. $\eta_t = \frac{1}{\sqrt{t}}$ is the step size, and $\nabla_{\boldsymbol{\theta}}\rho_{r_t}(\boldsymbol{\theta})$ is the gradient of $\rho_{r_t}(\boldsymbol{\theta})$:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\rho_{r_t}(\boldsymbol{\theta}) &\equiv \left[\frac{\partial\rho_{r_t}(\boldsymbol{\theta})}{\partial\theta^{(1)}}, \ldots, \frac{\partial\rho_{r_t}(\boldsymbol{\theta})}{\partial\theta^{(N)}}\right]^\top \\ &= \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{a}\in A} d_{\boldsymbol{\theta}}(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})(\nabla_{\boldsymbol{\theta}}\ln d_{\boldsymbol{\theta}}(\boldsymbol{s}) + \nabla_{\boldsymbol{\theta}}\ln\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})) \\ &\qquad\qquad \times r_t(\boldsymbol{s},\boldsymbol{a})\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}. \end{aligned} \tag{3.5}$$

In Eq.(3.5), the facts $\nabla_{\boldsymbol{\theta}}\ln d_{\boldsymbol{\theta}}(\boldsymbol{s}) = \frac{\nabla_{\boldsymbol{\theta}} d_{\boldsymbol{\theta}}(\boldsymbol{s})}{d_{\boldsymbol{\theta}}(\boldsymbol{s})}$ and $\nabla_{\boldsymbol{\theta}}\ln\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) = \frac{\nabla_{\boldsymbol{\theta}}\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})}{\pi(\boldsymbol{a}|\boldsymbol{s};\theta)}$ are used. If it is time-consuming to obtain the exact stationary state distribution, gradients estimated by a reinforcement learning algorithm may be used instead in practice. Since the transition and reward functions are known to the agent, it is straightforward to estimate the gradient efficiently by using a reinforcement learning technique (e.g., REINFORCE and policy gradients with parameter based exploration) (Sutton and Barto, 1998; Williams, 1992; Sehnke et al., 2010). Furthermore, some reinforcement learning techniques provided a convergence guarantee for the gradient estimation. Especially in the REINFORCE algorithm, the gradient is approximated by the empirical average value $\nabla_{\boldsymbol{\theta}}\bar{\rho}_t(\boldsymbol{\theta})$ after sufficient trajectories are collected as

$$\nabla_{\boldsymbol{\theta}}\bar{\rho}_{r_t}(\boldsymbol{\theta}) = \frac{1}{|H|}\sum_{n=1}^{|H|}\sum_{i=1}^{L}\nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{a}_i|\boldsymbol{s}_i;\boldsymbol{\theta})R(\boldsymbol{h}_n),$$

where $\boldsymbol{h}_n$ is a roll-out sample denoted as $\boldsymbol{h}_n = [\boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{s}_L, \boldsymbol{a}_L]$, the set of collected trajectories with length L is $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{|H|}\}$, and $R(\boldsymbol{h}_n)$ is the average reward obtained by trajectory $\boldsymbol{h}_n$. With theoretical guarantee, the REINFORCE algorithm has been shown to converge to the true gradient as $|H|$ and L tend to infinity. In the following analysis, we ignore the approximation error since it could be arbitrarily small by collecting a large enough number of samples.

When the reward function does not changed over time, the OPG algorithm is reduced to the ordinary policy gradient algorithm (Williams, 1992), which is an efficient and natural algorithm for continuous state and action MDPs. The OPG algorithm can also be regarded as an extension of the *online gradient descend* algorithm (Zinkevich, 2003), which maximizes $\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t)$, not $\mathbb{E}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)|\mathcal{A}\right]$. As we showed in the definition of $\rho_{r_t}(\boldsymbol{\theta}_t)$, the stationary state distribution $d_{\boldsymbol{\theta}_t}(\boldsymbol{s})$ of policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)$ is used, which is different from the state distribution $d_{\mathcal{A},t}(\boldsymbol{s})$ used in $\mathbb{E}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)|\mathcal{A}\right]$. As we will prove in Section 3.3, the regret bound of the OPG algorithm is $O(\sqrt{T})$ under a certain concavity assumption and $O(\log T)$ under a strong concavity assumption on the expected average reward function. Unlike previous works (Even-Dar et al., 2003, 2009; Yu et al., 2009; Neu et al., 2010b), these bounds do not depend on the cardinality of state and action spaces. Therefore, the OPG algorithm would be suitable for handling continuous state and action online MDPs.

3.2.3 OPG with Bandit Feedback

Here we extend the OPG algorithm to the *bandit feedback* scenario, where the *entire* reward function is not available, but only the value of the reward function for the current state and action is observed:

$$\boldsymbol{s}_1, \boldsymbol{a}_1, r_1(\boldsymbol{s}_1, \boldsymbol{a}_1), \ldots, \boldsymbol{s}_t, \boldsymbol{a}_t, r_t(\boldsymbol{s}_t, \boldsymbol{a}_t).$$

Due to lack of the entire reward function, we replace reward function r_t in the OPG algorithm with a random reward function given by

$$\hat{r}_t(\boldsymbol{s}, \boldsymbol{a}) = \frac{r_t(\boldsymbol{s}, \boldsymbol{a})}{d_{\mathcal{A},t}(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)} \delta(\boldsymbol{s}_t = \boldsymbol{s}, \boldsymbol{a}_t = \boldsymbol{a}). \tag{3.6}$$

Note that the above reward function is an unbiased estimator of $r_t(\boldsymbol{s}, \boldsymbol{a})$ for all $t = 1, \ldots, T$ (Yu et al., 2009):

$$\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})}[\hat{r}_t(\boldsymbol{s}, \boldsymbol{a})|\mathcal{A}] = r_t(\boldsymbol{s}, \boldsymbol{a}), \forall \boldsymbol{s} \in S, \boldsymbol{a} \in A.$$

In above equation, $\mathbb{E}_{p(\boldsymbol{s}_t,\boldsymbol{a}_t)}[\cdot|\mathcal{A}]$ denotes the expectation over the joint state-action distribution $p_t(\boldsymbol{s}, \boldsymbol{a})$ by the policies picked by algorithm $\mathcal{A}$ at time step t, where $p_t(\boldsymbol{s}, \boldsymbol{a}) = d_{\mathcal{A},t}(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)$. By the definition $\rho_{r_t}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}\sim d_{\boldsymbol{\theta}}(\boldsymbol{s}),\boldsymbol{a}\sim\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})}[r_t(\boldsymbol{s}, \boldsymbol{a})]$, the estimated expected average reward function satisfies

$$\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})}\left[\hat{\rho}_{r_t}(\boldsymbol{\theta})|\mathcal{A}\right] = \rho_{r_t}(\boldsymbol{\theta}),$$

where

$$\hat{\rho}_{r_t}(\boldsymbol{\theta}) = \int_{\boldsymbol{s}\in S} d_{\boldsymbol{\theta}}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} \hat{r}_t(\boldsymbol{s},\boldsymbol{a})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}.$$

The gradient of $\hat{\rho}_{r_t}(\boldsymbol{\theta})$ with respect to the parameter $\boldsymbol{\theta}$ can be obtained by passing the derivative through the integral as

$$\begin{aligned}\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})}\left[\frac{\partial \hat{\rho}_{r_t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|\mathcal{A}\right] &= \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{a}\in A} d_{\mathcal{A},t}(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)\frac{\partial \hat{\rho}_{r_t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}\\ &= \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{a}\in A}\left(\frac{\partial \log d_\theta(\boldsymbol{s})}{\partial \boldsymbol{\theta}} + \frac{\partial \log \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\\ &\quad \times d_\theta(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})r_t(\boldsymbol{s},\boldsymbol{a})\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}\\ &= \frac{\partial \rho_{r_t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}.\end{aligned}$$

As the above equation shows, we replaced the gradient of the expected average reward function $\frac{\partial \rho_{r_t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ in Eq.(3.4) with its unbiased estimator $\frac{\partial \hat{\rho}_{r_t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$.

As will be proved in Section 3.4, the regret bound of the OPG method with bandit feedback is still $O(\sqrt{T})$, although the bound is looser than that in the full-feedback case. If it is not possible to calculate the state distribution directly, its estimate obtained by reinforcement learning may be employed in practice (Ng et al., 1999).

3.3 Regret Analysis with Full Feedback

In this section, we present the main theorem of the proposed online policy gradient algorithm with full information feedback.

3.3.1 Assumptions

First, we introduce the assumptions required in the proofs. Some assumptions have already been used in related works for discrete state and action MDPs, and we extend them to continuous state and action MDPs.

Assumption 4. *There exists a positive number τ, such that for two arbitrary distributions d and d' over S and for every policy parameter $\boldsymbol{\theta} \in \Theta$,*

$$\int_{\boldsymbol{s}\in S}\int_{\boldsymbol{s}'\in S} |d(\boldsymbol{s}) - d'(\boldsymbol{s})|p(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta})\mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \leq e^{-1/\tau}\int_{\boldsymbol{s}\in S}|d(\boldsymbol{s}) - d'(\boldsymbol{s})|\mathrm{d}\boldsymbol{s},$$

where

$$p(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta}) = \int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathrm{d}\boldsymbol{a}.$$

τ *is called the* mixing time *(Even-Dar et al., 2003, 2009).*

Assumption 5. *There exists a positive constant C_1 depending on the specific policy model π, such that for two arbitrary policy parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ and for every $\boldsymbol{s} \in S$,*

$$\int_{\boldsymbol{a}\in A} |\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) - \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}')|\mathrm{d}\boldsymbol{a} \leq C_1\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1,$$

where $\|\cdot\|_1$ denotes the L_1 norm.

The Gaussian policy is a common choice in continuous state and action MDPs. Below, we consider the Gaussian policy with mean $\mu(\boldsymbol{s}) = \boldsymbol{\theta}^\top\phi(\boldsymbol{s})$ and standard deviation σ, where $\boldsymbol{\theta}$ is the policy parameter and $\phi(\boldsymbol{s}) : S \to \mathbb{R}^N$ is the basis function. The KL-divergence between these two policies is given by

$$\begin{aligned} D(p(\cdot|s;\boldsymbol{\theta})||p(\cdot|s;\boldsymbol{\theta'})) &= \int_{a\in A} \mathcal{N}_{\theta,\sigma}(a)\left\{\log\mathcal{N}_{\theta,\sigma}(a) - \log\mathcal{N}_{\theta',\sigma}(a)\right\}\mathrm{d}a \\ &= \int_{a\in A} \mathcal{N}_{\theta,\sigma}(a)\left\{\frac{1}{2\sigma^2}\left(-(a-\boldsymbol{\theta}^\top\phi(\boldsymbol{s}))^2 + (a-\boldsymbol{\theta}'^\top\phi(\boldsymbol{s}))^2\right)\right\}\mathrm{d}a \\ &\leq \frac{\|\phi(\boldsymbol{s})\|_\infty^2}{2\sigma^2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1^2. \end{aligned}$$

By Pinsker's inequality, the following inequality holds:

$$\|p(\cdot|s,\boldsymbol{\theta}) - p(\cdot|s,\boldsymbol{\theta}')\|_1 \leq \frac{\|\phi(\boldsymbol{s})\|_\infty}{\sigma}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1. \tag{3.7}$$

This implies that the Gaussian policy model satisfies Assumption 5 with $C_1 = \frac{\Phi}{\sigma}$, where $\|\phi(\boldsymbol{s})\|_\infty \leq \Phi, \forall \boldsymbol{s} \in S$. Note that we do not specify any policy model in the analysis, and therefore the following theoretical analysis is valid for other stochastic policy models as long as the assumptions are satisfied.

Assumption 6. *All the reward functions in online MDPs are bounded. For simplicity, we assume that the reward functions satisfy*

$$r_t(\boldsymbol{s},\boldsymbol{a}) \in [0,1], \forall \boldsymbol{s} \in S, \forall \boldsymbol{a} \in A, \forall t = 1,\ldots,T.$$

Assumption 7. *For all $t = 1, \dots, T$, the second derivative of the expected average reward function satisfies*

$$\nabla^2_\theta \rho_{r_t}(\boldsymbol{\theta}) \leq 0, \tag{3.8}$$

where $\boldsymbol{\theta} \in \Theta$ and Θ is the parameter set which is convex and compact.

This assumption means that the expected average reward function is concave, which is currently our sufficient condition to guarantee the $O(\sqrt{T})$-regret bound for the OPG algorithm. This assumption can be relaxed to locally concave expected average reward functions, where all the results still hold locally. More specifically the standard policy gradient algorithm (Sutton and Barto, 1998; Peters and Schaal, 2006) has been shown to converge to a local optimal solution, and we use a local optimal policy as the baseline in the definition of the regret instead of the global optimal solution.

3.3.2 Regret Bound with Concavity

In this section, we present our main result on the regret bound of the OPG algorithm under Assumption 7.

Theorem 3.1. *The regret against the best offline policy of the OPG algorithm is bounded as*

$$L_{\mathcal{A}}(T) \leq \sqrt{T}\frac{F^2}{2} + \sqrt{T}C_2N + 2\sqrt{T}\tau^2 C_1 C_2 N + 4\tau,$$

where F is the diameter of Θ and $C_2 = \frac{2C_1 - C_1 e^{-1/\tau}}{1-e^{-1/\tau}}$.

Note that the constant C_1 depends on the specific policy model involved which is claimed in Assumption 5.

To prove the above theorem, we decompose the regret in the same way as the previous work (Even-Dar et al., 2003, 2009; Neu et al., 2010a,b):

$$\begin{aligned} L_{\mathcal{A}}(T) =& R_{\boldsymbol{\theta}^*}(T) - R_{\mathcal{A}}(T) \\ \leq& \left(R_{\boldsymbol{\theta}^*}(T) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*) \right) + \left(\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t) \right) \\ &+ \left(\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t) - R_{\mathcal{A}}(T) \right). \end{aligned} \tag{3.9}$$

In the OPG method, $\rho_{r_t}(\boldsymbol{\theta})$ is used for optimization, and the sum of the expected average reward functions $\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*)$ is calculated based on the stationary state distribution $d_{\boldsymbol{\theta}^*}(\boldsymbol{s})$ of the policy parameterized by $\boldsymbol{\theta}^*$. However, the sum of the expected rewards $R_{\boldsymbol{\theta}^*}(T)$is calculated by $d_{\boldsymbol{\theta},t}(\boldsymbol{s})$, which is the state distribution at time step t following policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}^*)$. A similar argument can be obtained for $\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t)$ and $R_{\mathcal{A}}(T)$. These differences affect the first and third terms of the decomposed regret (3.9).

Below, we bound each of the three terms in Lemma 3.2, Lemma 3.3, and Lemma 3.4, which are proved in Appendix 3.6.1, Appendix 3.6.2, and Appendix 3.6.3, respectively.

Lemma 3.2. *The difference between the return and the expected average reward function of the best offline policy parameter satisfies*

$$\left| R_{\boldsymbol{\theta}^*}(T) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*) \right| \leq 2\tau.$$

The first term has already been analyzed for discrete state and action online MDPs in Even-Dar et al. (2003, 2009), Neu et al. (2014), and Dick et al. (2014), and we extended it to continuous state and action spaces in Lemma 3.2.

Lemma 3.3. *The expected average reward function satisfies*

$$\left| \sum_{t=1}^{T} (\rho_{r_t}(\boldsymbol{\theta}^*) - \rho_{r_t}(\boldsymbol{\theta}_t)) \right| \leq \sqrt{T}\frac{F^2}{2} + \sqrt{T} C_2 N.$$

Lemma 3.3 is obtained by using the result of Zinkevich (2003).

Lemma 3.4. *The difference between the return and the expected average reward function of* $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t), \forall t = 1, \ldots, T$ *given by the OPG algorithm* $\mathcal{A}$ *satisfies*

$$\left| R_{\mathcal{A}}(T) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t) \right| \leq 2\tau^2 C_1 C_2 N \sqrt{T} + 2\tau.$$

Lemma 3.4 is similar to Lemma 5.2 in Even-Dar et al. (2009), but our bound does not depend on the cardinality of state and action spaces.

Combining Lemma 3.2, Lemma 3.3, and Lemma 3.4, we can immediately obtain Theorem 3.1.

3.3.3 Regret Bound under Strong Concavity

Next we derive a sharper regret bound for the OPG algorithm under a strong concavity assumption.

Theorem 3.1 shows the theoretical guarantee of the OPG algorithm with the concave assumption. If the expected reward function is strongly concave, i.e.,

$$\nabla_\theta^2 \rho_{r_t} \leq -H I_N, \tag{3.10}$$

where H is a positive constant and I_N is the $N \times N$ identity matrix, we have following theorem.

Theorem 3.5. *The regret against the best offline policy of the OPG algorithm is bounded as*

$$L_{\mathcal{A}}(T) \leq \frac{C_2^2 N^2}{2H}(1 + \log T) + \frac{2\tau^2 C_1 C_2 N}{H} \log T + 4\tau,$$

with step size $\eta_t = \frac{1}{Ht}$.

In above theorem, $C_2 = \frac{2C_1 - C_1 e^{-1/\tau}}{1 - e^{-1/\tau}}$, where C_1 depends on the specific policy model. We again consider the same decomposition as Eq.(3.9), and the first term of the regret bound is exactly the same as Lemma 3.2.

The second term is bounded by the following proposition given the strong concavity assumption (3.10) and step size $\eta_t = \frac{1}{Ht}$:

Proposition 3.6.

$$\sum_{t=1}^{T} (\rho_{r_t}(\boldsymbol{\theta}^*) - \rho_{r_t}(\boldsymbol{\theta}_t)) \leq \frac{C_2^2 N^2}{2H}(1 + \log T).$$

The proof of Proposition 3.6 is given in Appendix 3.6.4, which follow the same line as Hazan et al. (2007).

From the proof of Lemma 3.4, the bound of the third term with the strong concavity assumption (3.10) is given by following proposition.

Proposition 3.7.

$$\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t) - R_{\mathcal{A}}(T) \leq \frac{2\tau^2 C_1 C_2 N}{H} \log T + 2\tau. \tag{3.11}$$

The result of Proposition 3.7 is obtained by following the same line as the proof of Lemma 3.4 with a different step size. Combining Lemma 3.2, Proposition 3.6, and Proposition 3.7, we can obtain Theorem 3.5.

3.4 Regret Analysis with Bandit Feedback

In this section, we prove a regret bound for the OPG algorithm in the bandit-feedback case.

Suppose that there exist $\xi > 0$ and $\epsilon > 0$ such that the policy and the state distribution satisfy

$$\begin{aligned}\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t) &\geq \xi, \forall \boldsymbol{s} \in S, \forall \boldsymbol{a} \in A, \forall t = 1, \ldots, T,\\ d_{\mathcal{A},t}(\boldsymbol{s}) &\geq \epsilon, \forall \boldsymbol{s} \in S, \forall t = 1, \ldots, T.\end{aligned}$$

Note that the above assumptions yield the state and action spaces to be compact, where the Gaussian policy cannot be used directly.

Then we have the following theorem:

Theorem 3.8. *The regret of the OPG algorithm with bandit feedback is*

$$\begin{aligned}L_{\mathcal{A}}(T) &= R_{\boldsymbol{\theta}^*}(T) - R_{\mathcal{A}}(T)\\ &\leq 4\tau + \frac{F^2}{2}\sqrt{T} + (C_3 + C_4)N\sqrt{T}\\ &\quad + 2\tau^2(C_1C_3N + C_1C_4N)\sqrt{T},\end{aligned}$$

where $C_3 = \frac{C_1}{\epsilon(1-e^{-1/\tau})}$, $C_4 = \frac{C_1}{\xi\epsilon}$, and C_1 depends on the specific policy model as Assumption 5.

Theorem 3.8 can be proved by extending the proof of Theorem 3.1 as follows.

The same regret decomposition as Eq.(3.9) is still possible in the bandit-feedback setting. The first term can be bounded in the same way as the full-information case, i.e., Lemma 3.2 still holds. However, the bounds for the second and third terms, originally given in Lemma 3.3 and Lemma 3.4, should be modified as follows:

Lemma 3.9. *The expected average reward function given by the online policy gradient algorithm with bandit feedback satisfies*

$$\left|\sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*) - \rho_{r_t}(\boldsymbol{\theta}_t)\right| \leq \frac{F^2}{2}\sqrt{T} + (C_3 + C_4)N\sqrt{T}.$$

The bound of the second part is still $O(\sqrt{T})$, but it is looser than the bound in the full-information scenario which is caused by the estimated gradient of the expected average reward function.

Lemma 3.10. *The third term of the regret of the online policy gradient algorithm with bandit feedback is bounded as*

$$\left| R_{\mathcal{A}}(T) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}_t) \right| \leq 2\tau^2 (C_1 C_3 N + C_1 C_4 N)\sqrt{T} + 2\tau.$$

Proofs of Lemma 3.9 and Lemma 3.10 are given in Appendix 3.6.7. From these lemmas, we can immediately obtain Theorem 3.8.

3.5 Experiments

In this section, we illustrate the behavior of the OPG algorithm through experiments.

3.5.1 Target Tracking

The task is to let an agent track an abruptly moving target located in one-dimensional real space $S = \mathbb{R}$. By abruptly, we mean that the target agent could jump from point to point. At the current time step, we cannot predict the target positions in the future. The action space is also one-dimensional real space $A = \mathbb{R}$, and we can change the position of the agent as $s' = s + a$. The reward function is given by evaluating the distance between the agent and target as

$$r_t(s, a) = e^{-\frac{1}{2}(s - \mathrm{tar}(t))^2 - \frac{1}{2}a^2}, \tag{3.12}$$

where $\mathrm{tar}(t) \in [-3, 3]$ denotes the position of the target at time step t. The mechanism for moving the target is set as the uniform distribution over the interval $[-3, 3]$.

We use the Gaussian policy with mean parameter $\mu = \theta \cdot s$ and standard deviation parameter $\sigma = 3$ in this experiment. From the standard argument (Peters and Schaal, 2006), the stationary state distribution is the Gaussian distribution with zero mean parameter and standard deviation parameter $\tilde{\sigma} = \frac{\sigma}{\sqrt{-\theta^2 - 2\theta}}, \theta \in$

$(-2, 0)$. Note that the parameter space is not closed in this experiment. When θ takes a value less than -1.99 or more than -0.01 during gradient update iterations, we project it back to -1.99 or -0.01, respectively. Then for all $t = 1, \ldots, T$, the expected average reward functions are given by

$$\begin{aligned}\rho_{r_t}(\theta) =& \int_{s \in S} \mathcal{N}_{0,\tilde{\sigma}}(s) \int_{a \in A} \mathcal{N}_{\mu,\sigma}(a) e^{-\frac{1}{2}(s-\mathrm{tar}(t))^2 - \frac{1}{2}a^2} \mathrm{d}a \mathrm{d}s \\ =& \frac{1}{\varpi} \exp\left(- \frac{\mathrm{tar}(t)^2(\varpi^2 - \tilde{\sigma}^2 - \sigma^2\tilde{\sigma}^2)}{2\varpi^2} \right),\end{aligned}$$

where $\varpi = \sqrt{1 + \sigma^2 + \tilde{\sigma}^2 + \sigma^2\tilde{\sigma}^2 + \tilde{\sigma}^2\theta^2}$. This implies that $\rho_{r_t}(\theta)$ is concave with respect to the parameter θ, and thus $\rho_{r_t}(\theta)$ satisfies Assumption 7 for all $t = 1, \ldots, T$. The analysis of concavity is presented in Appendix 3.6.9.

As a baseline method for comparison, we consider the MDP-E algorithm (Even-Dar et al., 2003, 2009), where the exponential weighted average algorithm is used as the best expert. Since MDP-E can handle only discrete states and actions, we discretize the state and action spaces. More specifically, the state space is discretized as

$$(-\infty, -6], (-6, -6 + c], (-6 + c, -6 + 2c], \ldots, (6, +\infty),$$

and the action space is discretized as

$$-6, -6 + c, -6 + 2c, \ldots, 6.$$

We consider the following 5 setups for c:

$$c = 12, 6, 2, 1, 0.5, 0.1.$$

In the experiment, the state distribution and the gradient are estimated by the policy gradient estimator REINFORCE introduced in Peters and Schaal (2006). $I = 20$ independent experiments are run with $T = 100$ time steps, and the average return $J(T)$ is used for evaluating the performance:

$$J(T) = \frac{1}{I} \sum_{i=1}^{I} \left[\sum_{t=1}^{T} r_t(s_t, a_t) \right].$$

The results are plotted in Figure 3.1, showing that the OPG algorithm works better than the MDP-E algorithm with the best discretization resolution. This illustrates the advantage of directly handling continuous state and action spaces without discretization. The MDP-E algorithm performs poorly when the discretization resolution is too small. Since the regret caused by the MDP-E algorithm increases as the cardinalities of state and action spaces increase. On the other hand, the performance of the MDP-E algorithm is limited when the discretization resolution is too large. Moreover, it is difficult to design the best discretization method without the knowledge of the target movement.

Figure 3.2 shows the average rewards and average regrets for full-information and bandit feedback cases, which substantiate the theoretical results.

Next, we set the state and action spaces as two-dimensional real spaces $S = \mathbb{R}^2, A = \mathbb{R}^2$. The target position $tar(t)$ is uniformly changing within $[-3, 3]^2$. The transition function is a linear function $\boldsymbol{s}' = \boldsymbol{s} + \boldsymbol{a}$. The reward function is given by evaluating the Euclidean distance between the agent and target as

$$r_t(\boldsymbol{s}, \boldsymbol{a}) = e^{-\frac{1}{2}\|\boldsymbol{s}-tar(t)\|_2^2 - \frac{1}{2}\|\boldsymbol{a}\|_2^2}.$$

For comparison, we discretize the state and action spaces with different resolutions. In Figure 3.3, we show the average returns obtained by the OPG algorithm and the MDP-E algorithm with different resolutions. This illustrates the OPG algorithm performs better than the MDP-E algorithm with the best discretization resolution.

3.5.2 Linear-quadratic Regulator

The *linear-quadratic regulator* (LQR) is a typical system, where the transition dynamics is linear and the reward function is quadratic. This system is instructive because we can compute the best offline parameter and the gradient directly (Peters and Schaal, 2006). Here, an online LQR system is simulated to illustrate the parameter update trajectory of the OPG algorithm.

Let state and action spaces be one-dimensional real space: $S = \mathbb{R}, A = \mathbb{R}$. The transitions are deterministically performed as

$$s' = s + a.$$

The reward function is defined as

$$r_t(s, a) = -\frac{1}{2}Q_t s^2 - \frac{1}{2}R_t a^2,$$

where $Q_t \in \mathbb{R}$ and $R_t \in \mathbb{R}$ are chosen from $\{1, \dots, 10\}$ uniformly at time step $t = 10, 20, 30, \dots, 10000$ [1]. Thus, the reward function is changing abruptly.

We use the Gaussian policy with mean parameter $\mu = \theta \cdot s$ and standard deviation parameter $\sigma = 0.1$ and $\sigma = 1$ in full information and bandit feedback experiments, respectively. The best offline parameter is given by $\theta^* = -0.92$, and the initial parameter for the OPG algorithm is drawn uniformly at random.

From the standard argument (Peters and Schaal, 2006), the expected average reward function of the above LQR system is given by

$$\rho_{r_t}(\theta) = -\frac{1}{2}(R_t + P_t)\sigma^2,$$

where P_t is the positive definite solution of the modified Ricatti equation $P_t = Q_t + P_t + 2\theta P_t + \theta^2 P_t + \theta^2 R_t$. Then the second order derivative of $\rho_{r_t}(\theta)$ is given by

$$\frac{\partial^2 \rho_{r_t}(\theta)}{\partial \theta^2} = \frac{\sigma^2 Q_t(6\theta^2 + 12\theta + 8) - 4\sigma^2\theta^3 R_t}{2(2\theta + \theta^2)^3}.$$

Given the fact that P is the positive definite solution which yields $-2 < \theta < 0$, we can obtain $\frac{\partial^2 \rho_{r_t}(\theta)}{\partial \theta^2} \leq 0$. This means that the expected average reward function of the target LQR system is always concave with respect to the policy parameter.

In the top graph of Figure 3.4, a parameter update trajectory of OPG with full information in the online LQR problem is plotted by the solid line, and the best offline parameter is denoted by the dashed line. This shows that the OPG solution quickly approaches the best offline parameter.

Next, we also include the Gaussian standard deviation σ in the policy parameter, i.e., $\boldsymbol{\theta} = (\mu, \sigma)^\top$. When σ takes a value less than 0.01 during gradient update iterations, we project it back to 0.01. A parameter update trajectory is plotted in the bottom graph of Figure 3.4, showing again that the OPG solution quickly approaches the best offline parameter value.

In the top graph of Figure 3.5, the solid line shows the trajectory of the OPG algorithm with bandit feedback in the online LQR system simulation. The result validates that the OPG solution converges to the best offline parameter with a slightly slower speed compared with the full information result.

[1] The reward function is not bounded, which violates Assumption 6. However, it is interesting to illustrate that the parameter updated by the OPG algorithm still converges to the best offline parameter.

The parameter trajectory is shown in the bottom graph of Figure 3.5 when the standard deviation σ is included in the parameter. The OPG solution still approaches the best offline mean parameter as we expect.

3.6 Proofs of Theorems

In this section, we present the proofs of all the theorems involved in this chapter.

3.6.1 Proof of Lemma 3.2

The following proposition holds, which can be obtained by recursively using Assumption 4:

Proposition 3.11. *For any policy parameter $\boldsymbol{\theta}$, the state distribution $d_{\boldsymbol{\theta},t}$ at time t and stationary state distribution $d_{\boldsymbol{\theta}}$ satisfy*

$$\int_{\boldsymbol{s}\in S} |d_{\theta,t}(\boldsymbol{s}) - d_{\theta}(\boldsymbol{s})| \mathrm{d}s \leq 2e^{-t/\tau}.$$

Then the first part of the regret bound could be bounded as

$$\begin{aligned}\left| R_{\boldsymbol{\theta}^*}(T) - \sum_{t=1}^{T} \rho_{r_t}(\boldsymbol{\theta}^*) \right| &= \left| \sum_{t=1}^{T} \left[\int_{\boldsymbol{s}\in S} d_{\theta^*,t}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} r_t(\boldsymbol{s},\boldsymbol{a})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}^*) \mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{a} \right.\right. \\ &\qquad \left.\left. - \int_{\boldsymbol{s}\in S} d_{\theta^*}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} r_t(\boldsymbol{s},\boldsymbol{a})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}^*) \mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{a} \right] \right| \\ &\leq \sum_{t=1}^{T} \int_{s\in S} |d_{\theta^*,t}(\boldsymbol{s}) - d_{\theta^*}(\boldsymbol{s})| \, \mathrm{d}\boldsymbol{s} \\ &\leq 2\sum_{t=1}^{T} e^{-t/\tau} \\ &\leq 2\tau, \end{aligned}$$

where the second inequality can be obtained by Assumption 4.

3.6.2 Proof of Lemma 3.3

The following proposition is a continuous extension of Lemma 6.3 in (Even-Dar et al., 2009):

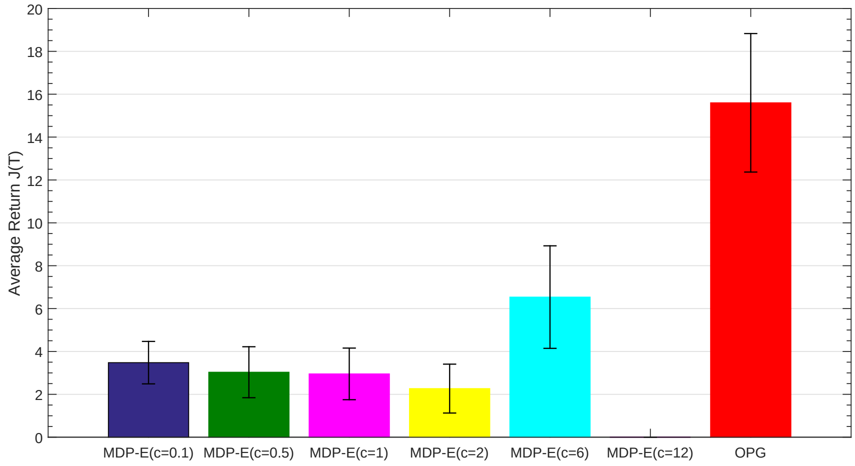


Figure 3.1: Average and standard deviation of returns of the OPG algorithm and the MDP-E algorithm with different discretization resolution c.

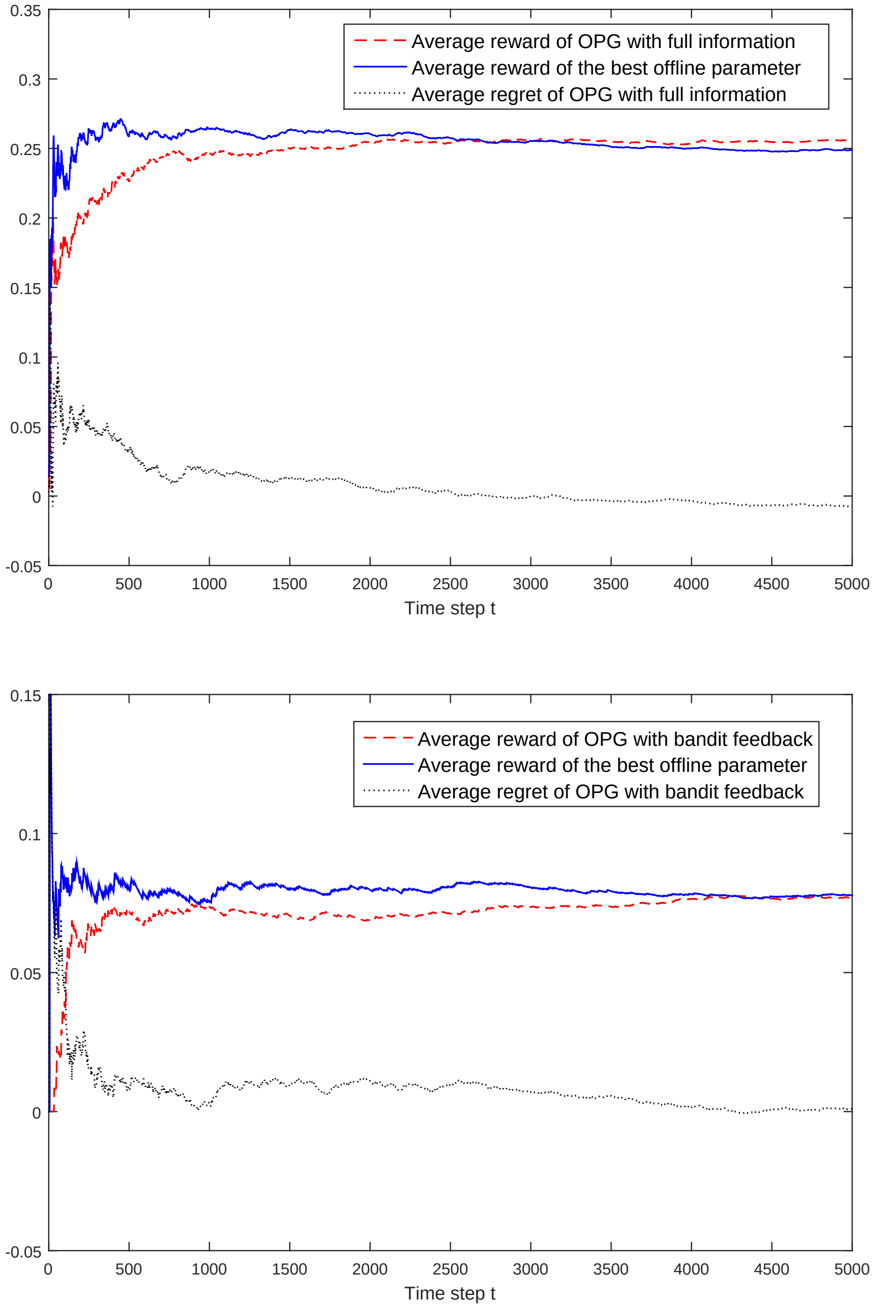


Figure 3.2: Average rewards and average regrets of the OPG algorithm with full information and bandit feedback.

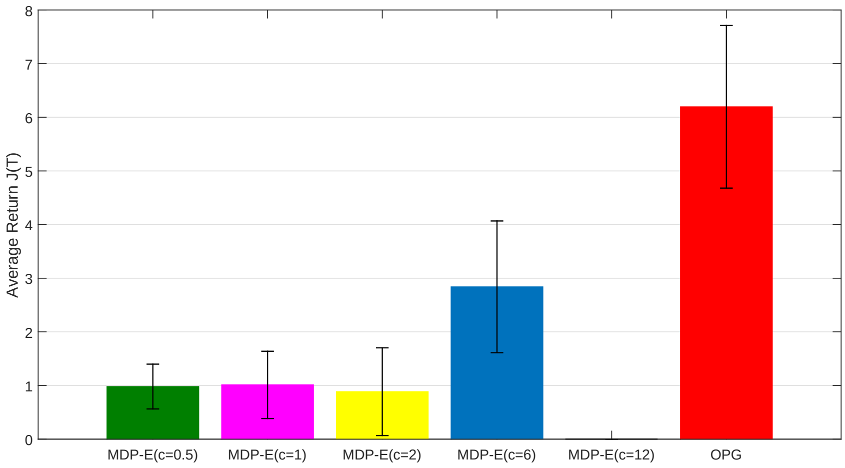


Figure 3.3: Average and standard deviation of returns of the OPG algorithm and the MDP-E algorithm with different discretization resolution c in 2-dimensional tracking experiment.

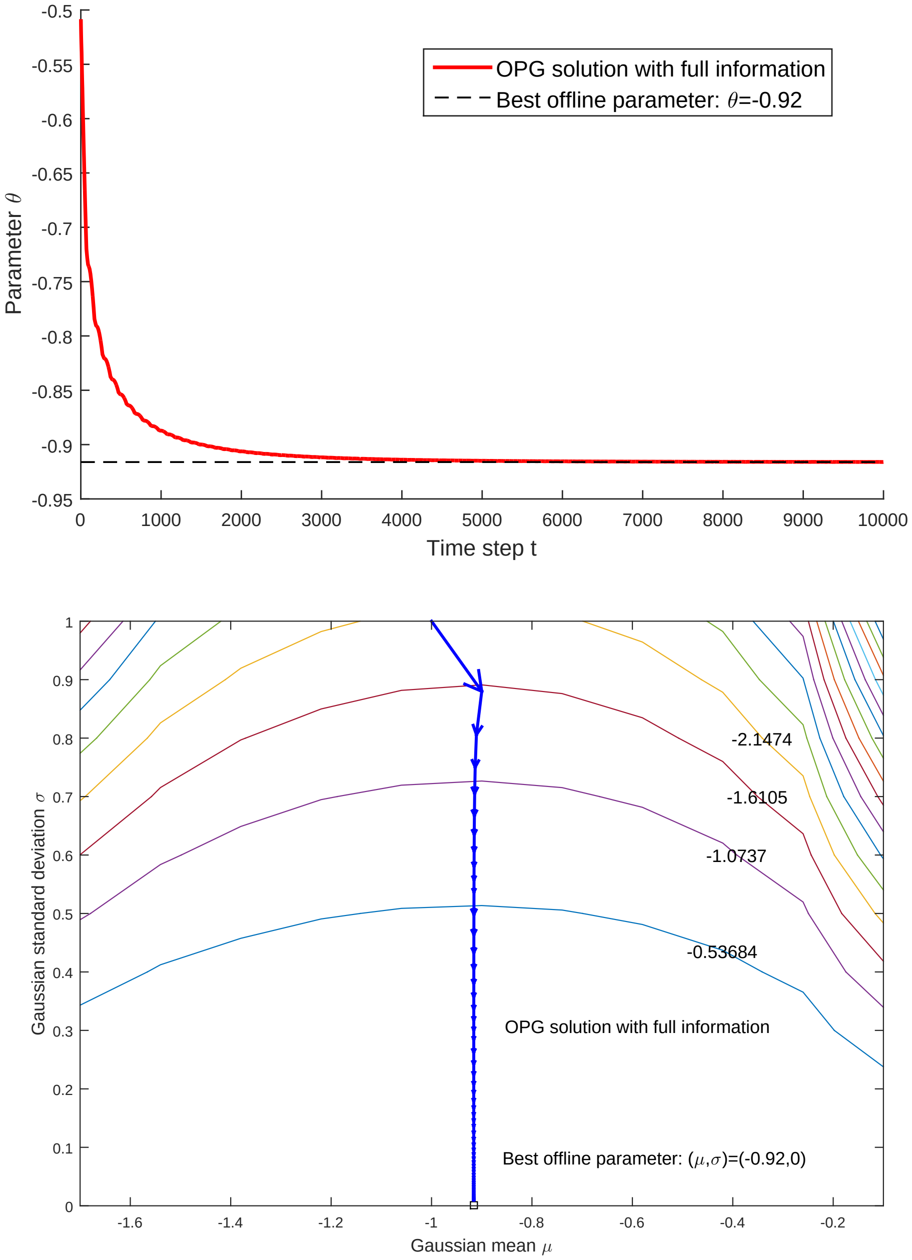


Figure 3.4: Trajectory of the OPG solution with full information and the best offline parameter.

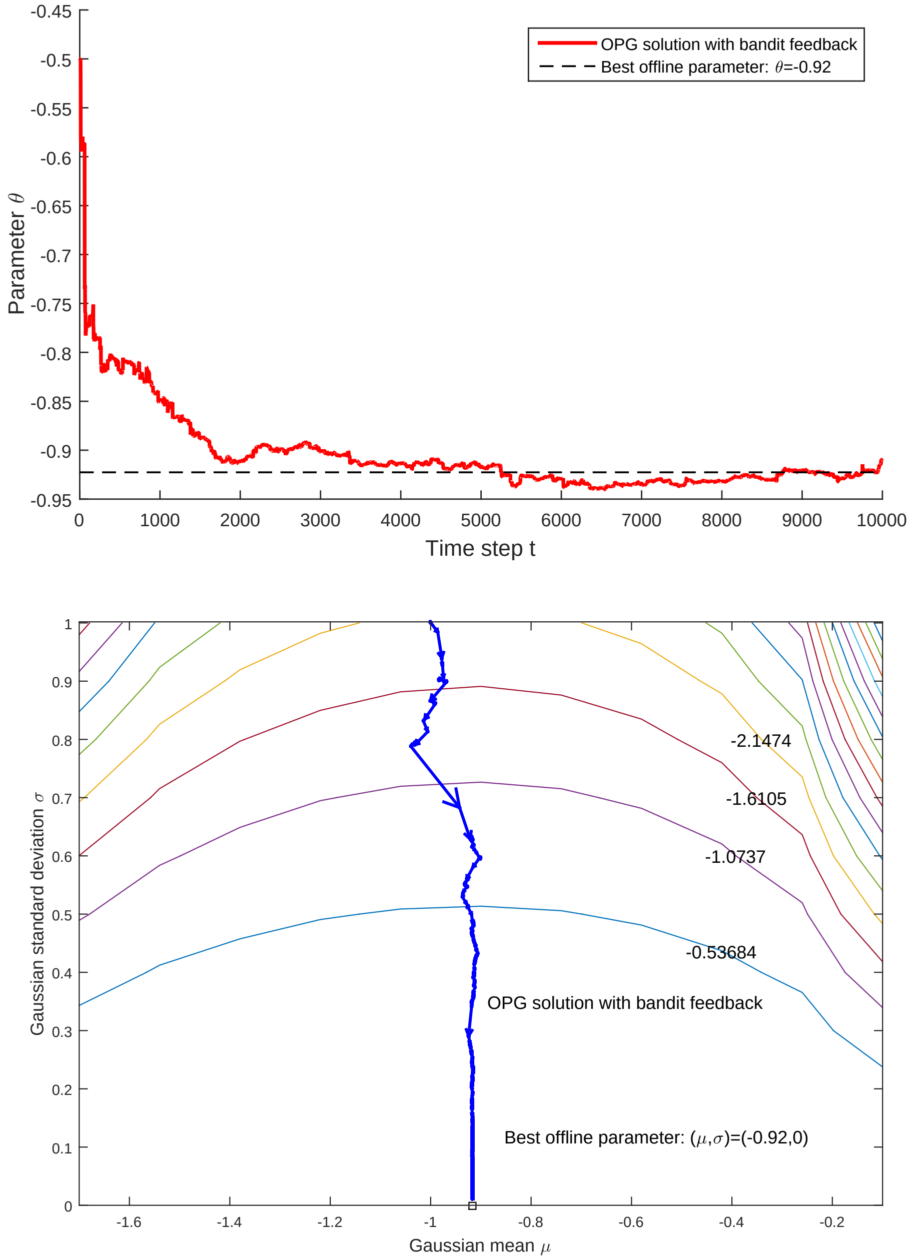


Figure 3.5: Trajectory of the OPG solution with bandit feedback and the best offline parameter.

Proposition 3.12. *For two policies with different parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, an arbitrary distribution d over S, and the constant $C_1 > 0$ given in Assumption 5, it holds that*

$$\int_{\boldsymbol{s}\in S} d(\boldsymbol{s}) \int_{\boldsymbol{s}'\in S} |p(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta}) - p(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta}')|\mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \leq C_1\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_1,$$

where

$$p(\boldsymbol{s}'|\boldsymbol{s};\boldsymbol{\theta}) = \int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta})p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathrm{d}\boldsymbol{a}.$$

Then we have the following proposition, which is proved in Appendix 3.6.5:

Proposition 3.13. *For all $t = 1,\ldots,T$, the expected average reward function $\rho_{r_t}(\boldsymbol{\theta})$ for two different parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ satisfies*

$$|\rho_{r_t}(\boldsymbol{\theta}) - \rho_{r_t}(\boldsymbol{\theta}')| \leq C_2\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_1.$$

From Proposition 3.13, we have the following proposition:

Proposition 3.14. *Let*

$$\begin{aligned}\boldsymbol{\theta} &= [\theta^{(1)},\ldots,\theta^{(i)},\ldots,\theta^{(N)}],\\ \boldsymbol{\theta}' &= [\theta^{(1)},\ldots,\theta^{(i)'},\ldots,\theta^{(N)}],\end{aligned}$$

and suppose that the expected average reward $\rho_{r_t}(\boldsymbol{\theta})$ for all $t = 1,\ldots,T$ is Lipschitz continuous with respect to each dimension $\theta^{(i)}$. Then we have

$$|\rho_{r_t}(\boldsymbol{\theta}) - \rho_{r_t}(\boldsymbol{\theta}')| \leq C_2|\theta^{(i)} - \theta^{(i)'}|, \forall i = 1,\ldots,N.$$

Form Proposition 3.14, we have the following proposition:

Proposition 3.15. *For all $t = 1,\ldots,T$, the partial derivative of expected average reward function $\rho_{r_t}(\boldsymbol{\theta})$ with respect to $\theta^{(i)}$ is bounded as*

$$\left|\frac{\partial\rho_{r_t}(\boldsymbol{\theta})}{\partial\theta^{(i)}}\right| \leq C_2, \forall i = 1,\ldots,N,$$

and $\|\nabla_{\boldsymbol{\theta}}\rho_{r_t}(\boldsymbol{\theta})\|_1 \leq NC_2$.

From Proposition 3.15, the result of online convex optimization (Zinkevich, 2003) is applicable to the current setup. More specifically we have

$$\sum_{t=1}^{T}(\rho_{r_t}(\boldsymbol{\theta}^*) - \rho_{r_t}(\boldsymbol{\theta}_t)) \leq \frac{F^2}{2}\sqrt{T} + C_2N\sqrt{T},$$

which concludes the proof.

3.6.3 Proof of Lemma 3.4

The following proposition holds, which can be obtained from Assumption 5 and

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\|_1 \leq \eta_t \|\nabla_{\boldsymbol{\theta}} \rho_{r_t}(\boldsymbol{\theta}_t)\|_1 \leq C_2 N \eta_t.$$

Proposition 3.16. *Consecutive policy parameters $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+1}$ given by the OPG algorithm satisfy*

$$\int_{\boldsymbol{a}\in A} |\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t) - \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_{t+1})| \mathrm{d}\boldsymbol{a} \leq C_1 C_2 N \eta_t.$$

From Proposition 3.12 and Proposition 3.16, we have the following proposition:

Proposition 3.17. *For consecutive policy parameters $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+1}$ given by the OPG algorithm and arbitrary transition probability density $p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$, it holds that*

$$\begin{aligned}&\int_{\boldsymbol{s}\in S} d(\boldsymbol{s}) \int_{\boldsymbol{s}'\in S} \int_{\boldsymbol{a}\in A} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\\ &\times |\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t) - \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_{t+1})| \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \leq C_1 C_2 N \eta_t.\end{aligned}$$

Then the following proposition holds, which is proved in Appendix 3.6.6 following the same line as Lemma 5.1 in Even-Dar et al. (2009):

Proposition 3.18. *The state distribution $d_{\mathcal{A},t}$ given by algorithm $\mathcal{A}$ and the stationary state distribution $d_{\boldsymbol{\theta}_t}$ of policy $\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)$ satisfy*

$$\int_{\boldsymbol{s}\in S} |d_{\boldsymbol{\theta}_t}(\boldsymbol{s}) - d_{\mathcal{A},t}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s} \leq 2\tau^2 \eta_{t-1} C_1 C_2 N + 2e^{-t/\tau}.$$

Although the original bound given in Even-Dar et al. (2003, 2009) depends on the cardinality of the action space, it is not the case in the current setup.

Then the third term of the decomposed regret (3.9) is expressed as

$$\begin{aligned}\left|R_{\mathcal{A}}(T)-\sum_{t=1}^{T}\rho_{r_t}(\boldsymbol{\theta}_t)\right| &= \left|\sum_{t=1}^{T}\int_{\boldsymbol{s}\in S}d_{\mathcal{A},t}(\boldsymbol{s})\int_{\boldsymbol{a}\in A}r_t(\boldsymbol{s},\boldsymbol{a})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}\right.\\ &\qquad\left.-\sum_{t=1}^{T}\int_{\boldsymbol{s}\in S}d_{\boldsymbol{\theta}_t}(\boldsymbol{s})\int_{\boldsymbol{a}\in A}r_t(\boldsymbol{s},\boldsymbol{a})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)\mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}\right|\\ &\le \sum_{t=1}^{T}\int_{\boldsymbol{s}\in S}|d_{\mathcal{A},t}(\boldsymbol{s})-d_{\pi_t}(\boldsymbol{s})|\mathrm{d}\boldsymbol{s}\\ &\le 2\tau^2C_1C_2N\sum_{t=1}^{T}\eta_t+2\sum_{t=1}^{T}e^{-t/\tau}\\ &\le 2\tau^2C_1C_2N\sqrt{T}+2\tau,\end{aligned}$$

which concludes the proof.

3.6.4 Proof of Proposition 3.6

The proof of Proposition 3.6 can be obtained from Hazan et al. (2007), i.e., by the Taylor approximation, the expected average reward function can be decomposed as

$$\begin{aligned}&\rho_{r_t}(\boldsymbol{\theta}^*)-\rho_{r_t}(\boldsymbol{\theta}_t)\\ &=\nabla_\theta\rho_{r_t}(\boldsymbol{\theta}_t)^\top(\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)+\frac{1}{2}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)^\top\nabla_\theta^2\rho_{r_t}(\boldsymbol{\xi}_t)(\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)\\ &\le\nabla_\theta\rho_{r_t}(\boldsymbol{\theta}_t)^\top(\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)-\frac{H}{2}\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_t\|^2,\end{aligned}\tag{3.13}$$

where $\boldsymbol{\xi}_t$ is some point between $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}_t$. The last inequality comes from the strong concavity assumption (3.10). Given the parameter updating rule,

$$\nabla_\theta\rho_{r_t}(\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)=\frac{1}{2\eta_t}\left((\boldsymbol{\theta}^*-\boldsymbol{\theta}_t)^2-(\boldsymbol{\theta}^*-\boldsymbol{\theta}_{t+1})^2\right)+\eta_t\|\nabla_\theta\rho_{r_t}(\boldsymbol{\theta}_t)\|^2,$$

summing up all T terms of (3.13) and setting $\eta_t=\frac{1}{Ht}$ yield

$$\begin{aligned}\sum_{t=1}^{T}(\rho_{r_t}(\boldsymbol{\theta}^*)-\rho_{r_t}(\boldsymbol{\theta}_t))&\le\sum_{t=1}^{T}\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}-H\right)\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_t\|^2+\|\nabla_t\rho_{r_t}(\boldsymbol{\theta}_t)\|^2\sum_{t=1}^{T}\eta_t\\ &\le\frac{C_2^2N^2}{2H}(1+\log T).\end{aligned}$$

3.6.5 Proof of Proposition 3.13

For two different parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, we have

$$\begin{aligned}
|\rho_{r_t}(\boldsymbol{\theta}) - \rho_{r_t}(\boldsymbol{\theta}')| &= \left| \int_{\boldsymbol{s}\in S} d_{\boldsymbol{\theta}}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s} \right. \\
&\qquad \left. - \int_{\boldsymbol{s}\in S} d_{\boldsymbol{\theta}'}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}') r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s} \right| \\
&\leq \int_{\boldsymbol{s}\in S} |d_{\boldsymbol{\theta}}(\boldsymbol{s}) - d_{\boldsymbol{\theta}'}(\boldsymbol{s})| \int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s} \\
&\qquad + \int_{\boldsymbol{s}\in S} d_{\boldsymbol{\theta}'}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} |\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) - \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}')| \, r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s}.
\end{aligned} \tag{3.14}$$

The first equation comes from Eq.(3.3), and the second inequality is obtained from the triangle inequality. Since Assumption 5 and Assumption 6 imply

$$\int_{\boldsymbol{s}\in S} d_{\boldsymbol{\theta}'}(\boldsymbol{s}) \int_{\boldsymbol{a}\in A} |\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) - \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}')| r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a}\mathrm{d}\boldsymbol{s} \leq C_1 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1,$$

and also

$$\int_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}) r_t(\boldsymbol{s},\boldsymbol{a}) \mathrm{d}\boldsymbol{a} \leq 1,$$

Eq.(3.14) can be written as

$$\begin{aligned}
|\rho_{r_t}(\boldsymbol{\theta}) - \rho_{r_t}(\boldsymbol{\theta}')| &\leq \int_{\boldsymbol{s}\in S} |d_{\boldsymbol{\theta}}(\boldsymbol{s}) - d_{\boldsymbol{\theta}'}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s} + C_1 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1 \\
&= \int_{\boldsymbol{s}\in S} \int_{\boldsymbol{s}'\in S} |d_{\boldsymbol{\theta}}(\boldsymbol{s}') p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}) - d_{\boldsymbol{\theta}'}(\boldsymbol{s}') p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}')| \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\qquad + C_1 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1 \\
&\leq \int_{\boldsymbol{s}\in S} \int_{\boldsymbol{s}'\in S} |d_{\boldsymbol{\theta}}(\boldsymbol{s}') p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}) - d_{\boldsymbol{\theta}'}(\boldsymbol{s}') p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta})| \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\qquad + \int_{\boldsymbol{s}\in S} \int_{\boldsymbol{s}'\in S} d_{\boldsymbol{\theta}'}(\boldsymbol{s}') |p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}) - p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}')| \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\qquad + C_1 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1 \\
&\leq e^{-1/\tau} \int_{\boldsymbol{s}\in S} |d_{\boldsymbol{\theta}}(\boldsymbol{s}) - d_{\boldsymbol{\theta}'}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s} + 2C_1 \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1.
\end{aligned}$$

The second equality comes from the definition of the stationary state distribution, and the third inequality can be obtained from the triangle inequality. The last inequality follows from Assumption 4 and Proposition 3.12. Thus, we have

$$|\rho_{r_t}(\boldsymbol{\theta}) - \rho_{r_t}(\boldsymbol{\theta}')| \leq \frac{2C_1 - C_1 e^{-1/\tau}}{1 - e^{-1/\tau}} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_1,$$

which concludes the proof.

3.6.6 Proof of Proposition 3.18

This proof is following the same line as Lemma 5.1 in Even-Dar et al. (2009).

$$\begin{aligned}
&\int_{\boldsymbol{s}\in S} |d_{\mathcal{A},k}(\boldsymbol{s}) - d_{\boldsymbol{\theta}_t}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s} \\
&= \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{s}'\in S} |d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_k) - d_{\boldsymbol{\theta}_t}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_t)|\, \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\leq \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{s}'\in S} |d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_t) - d_{\boldsymbol{\theta}_t}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_t)|\, \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\quad + \int_{\boldsymbol{s}\in S}\int_{\boldsymbol{s}'\in S} |d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_k) - d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_t)|\, \mathrm{d}\boldsymbol{s}'\mathrm{d}\boldsymbol{s} \\
&\leq e^{-1/\tau}\int_{\boldsymbol{s}\in S} |d_{\mathcal{A},k-1}(\boldsymbol{s}) - d_{\boldsymbol{\theta}_t}(\boldsymbol{s})|\, \mathrm{d}\boldsymbol{s} + 2(t-k)C_1C_2N\eta_{t-1}. \qquad (3.15)
\end{aligned}$$

The first equation comes from the definition of the stationary state distribution, and the second inequality can be obtained by the triangle inequality. The third inequality holds from Assumption 4 and

$$\begin{aligned}
&\int_{\boldsymbol{s}\in S}\int_{\boldsymbol{s}'\in S} |d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_k) - d_{\mathcal{A},k-1}(\boldsymbol{s}')p(\boldsymbol{s}|\boldsymbol{s}';\boldsymbol{\theta}_t)|\, \mathrm{d}\boldsymbol{s} \\
&\leq C_1\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_k\|_1 \\
&\leq C_1\sum_{i=k}^{t-1}\eta_i\|\nabla_{\boldsymbol{\theta}}\rho_i(\boldsymbol{\theta}_i)\|_1 \\
&\leq 2(t-k)C_1C_2N\eta_{t-1}.
\end{aligned}$$

Recursively using Eq.(3.15), we have

$$\begin{aligned}
\int_{\boldsymbol{s}\in S} |d_{\mathcal{A},t}(\boldsymbol{s}) - d_{\pi_t}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s} &\leq 2\sum_{k=2}^{t} e^{-(t-k)/\tau}(t-k)C_1C_2N\eta_{t-1} + 2e^{-t/\tau} \\
&\leq 2\tau^2C_1C_2N\eta_{t-1} + 2e^{-t/\tau},
\end{aligned}$$

which concludes the proof.

3.6.7 Proofs of Lemma 3.9 and Lemma 3.10

As we show in Section 3.4, an unbiased estimator of reward function is used for updating the parameter $\boldsymbol{\theta}$, we also show that the corresponding estimated gradient is unbiased which can be bounded by the following lemma, which is proved in Appendix 3.6.8.

Lemma 3.19. *The estimated gradient* $\nabla_{\boldsymbol{\theta}} \hat{\rho}_{r_t}(\boldsymbol{\theta})$ *satisfies*

$$\|\nabla_{\boldsymbol{\theta}} \hat{\rho}_{r_t}(\boldsymbol{\theta})\|_1 \leq C_3 N + C_4 N.$$

Following the same line with the proof of Lemma 3.1 in Flaxman et al. (2005), we firstly define the auxiliary functions for all $\boldsymbol{x} \in \Theta$ as

$$\varrho_t(\boldsymbol{x}) = \rho_{r_t}(\boldsymbol{x}) + \boldsymbol{x}^\top \kappa_t,$$

where $\kappa_t = \nabla_{\boldsymbol{\theta}} \hat{\rho_{r_t}}(\boldsymbol{\theta}_t) - \nabla_{\boldsymbol{\theta}} \rho_{r_t}(\boldsymbol{\theta}_t)$. Observed that

$$\nabla_{\boldsymbol{x}} \varrho_t(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}} \hat{\rho}_{r_t}(\boldsymbol{\theta}_t),$$

and the unbiased estimation satisfies

$$\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})} \left[\varrho_t(\boldsymbol{\theta}_t) | \mathcal{A}\right] = \rho_{r_t}(\boldsymbol{\theta}_t),$$

where the above equation follows from the fact $\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})}[\kappa_t|\mathcal{A}] = 0$, and $\mathbb{E}_{p_t(\boldsymbol{s},\boldsymbol{a})}[\boldsymbol{\theta}_t \kappa_t|\mathcal{A}] = 0$. Thus, we can obtain

$$\sum_{t=1}^{T} \left(\rho_{r_t}(\boldsymbol{\theta}^*) - \rho_{r_t}(\boldsymbol{\theta}_t)\right) \leq \frac{F^2}{2}\sqrt{T} + (C_3 + C_4) N \sqrt{T},$$

which concludes the proof of Lemma 3.9 by using the result of Lemma 3.19. Similarly, using Lemma 3.19 in the proof of Lemma 3.4, we obtain Lemma 3.10.

3.6.8 Proof of Lemma 3.19

The estimated gradient is expressed as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\hat{\rho}_{r_t}(\boldsymbol{\theta}_t) = &\int_{\boldsymbol{s}\in S}\int_{\boldsymbol{a}\in A} d_{\boldsymbol{\theta}_t}(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)\hat{r}_t(\boldsymbol{s},\boldsymbol{a})\\ &\times(\nabla_{\boldsymbol{\theta}}\ln d_{\boldsymbol{\theta}_t}(\boldsymbol{s}) + \nabla_{\boldsymbol{\theta}}\ln\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t))d\boldsymbol{s}d\boldsymbol{a}\\ =&\frac{\nabla_{\boldsymbol{\theta}}d_{\boldsymbol{\theta}_t}(\boldsymbol{s}_t)}{d_{\mathcal{A},t}(\boldsymbol{s}_t)}r_t(\boldsymbol{s}_t,\boldsymbol{a}_t)\\ &+\frac{d_{\boldsymbol{\theta}_t}(\boldsymbol{s}_t)}{d_{\mathcal{A},t}(\boldsymbol{s}_t)}\ln\nabla_{\boldsymbol{\theta}}\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)r_t(\boldsymbol{s}_t,\boldsymbol{a}_t).\end{aligned}$$

Consider the stationary distribution as a function of parameter $\boldsymbol{\theta}$ for all $\boldsymbol{s}\in S$, Then, from Proposition 3.13, the bound for the gradient of the stationary distribution is given by

$$|\nabla_{\boldsymbol{\theta}}d_{\boldsymbol{\theta}_t}(\boldsymbol{s})| \leq \frac{C_1N}{1-e^{-1/\tau}}, \forall\boldsymbol{s}\in S, \forall t=1,\dots,T.$$

Similarly, from Assumption 5, the bound for the gradient of policy π is given by

$$|\nabla_{\boldsymbol{\theta}}\ln\pi(\boldsymbol{a}|\boldsymbol{s};\boldsymbol{\theta}_t)| \leq \frac{C_1N}{\xi}, \forall\boldsymbol{s}\in S, \forall\boldsymbol{a}\in A, \forall t=1,\dots,T.$$

Then we have

$$\|\nabla_{\boldsymbol{\theta}}\hat{\rho}_{r_t}(\boldsymbol{\theta}_t)\|_1 \leq \frac{C_1N}{\epsilon(1-e^{-1/\tau})} + \frac{C_1N}{\epsilon\xi}, \forall t=1,\dots,T.$$

3.6.9 Concavity Analysis for Target Tracking

The reward function in the target tracking experiment is defined as

$$r_t(s,a) = e^{-\frac{1}{2}(s-\mathrm{tar}(t))^2-\frac{1}{2}a^2}, \forall t=1,\dots,T.$$

Then for all $t=1,\dots,T$, the expected average reward function are given by

$$\begin{aligned}\rho_{r_t}(\theta) =&\int_{s\in S}\mathcal{N}_{0,\tilde{\sigma}}(s)\int_{a\in A}\mathcal{N}_{\mu,\sigma}(a)e^{-\frac{1}{2}(s-\mathrm{tar}(t))^2-\frac{1}{2}a^2}\mathrm{d}a\mathrm{d}s\\ =&\frac{1}{\varpi}\exp\left(-\frac{\mathrm{tar}(t)^2(\varpi^2-\tilde{\sigma}^2-\sigma^2\tilde{\sigma}^2)}{2\varpi^2}\right),\end{aligned}$$

where $\varpi = \sqrt{1 + \sigma^2 + \tilde{\sigma}^2 + \sigma^2\tilde{\sigma}^2 + \tilde{\sigma}^2\theta^2}$ and $\tilde{\sigma} = \frac{\sigma}{\sqrt{-\theta^2 - 2\theta}}$. For verifying the concavity of $\rho_{r_t}(\theta)$, we obtain the derivative of $\rho_{r_t}(\theta)$ with respect to θ by plugging in $\sigma = 3$ as

$$\begin{aligned}\frac{\partial \rho_{r_t}(\theta)}{\partial \theta} =& \sqrt{\frac{-\theta^2 - 2\theta}{-\theta^2 - 20\theta + 90}} \exp\left(-\frac{t^2}{2} \cdot \frac{-\theta^2 - 20\theta}{-\theta^2 - 20\theta + 90}\right) \\ & \times \left[-\mathrm{tar}(t)^2 \frac{-90(\theta + 10)}{(-\theta^2 - 20\theta + 90)^2} - \frac{-9\theta^2 + 90\theta + 90}{(-\theta^2 - 20\theta + 90)(-\theta^2 - 2\theta)}\right].\end{aligned}$$

Observed that $\frac{\partial \rho_{r_t}(\theta)}{\partial \theta}$ is monotonically non-increasing as shown in Figure 3.6. Thus, the defined expected average reward functions $\rho_{r_t}(\theta), \forall t = 1, \ldots, T$ are concave with respect to the parameter θ.

3.7 Summary

In this chapter, we proposed an online policy gradient method for continuous state and action online MDPs, and showed that the regret of the proposed method is $O(\sqrt{T})$ under a certain concavity assumption on the expected average reward function. A notable fact is that the regret bound does not depend on the cardinality of state and action spaces, which makes the proposed algorithm suitable in handling continuous states and actions. We further extended our method to the bandit-feedback scenario, and showed that the regret of the extended method is still $O(\sqrt{T})$. Furthermore, we also established the $O(\log T)$ regret bound under a strong concavity assumption for the full information setup. Through experiments, we illustrated that directly handling continuous state and action spaces by the proposed method is more advantageous than discretizing them and applying an existing method.

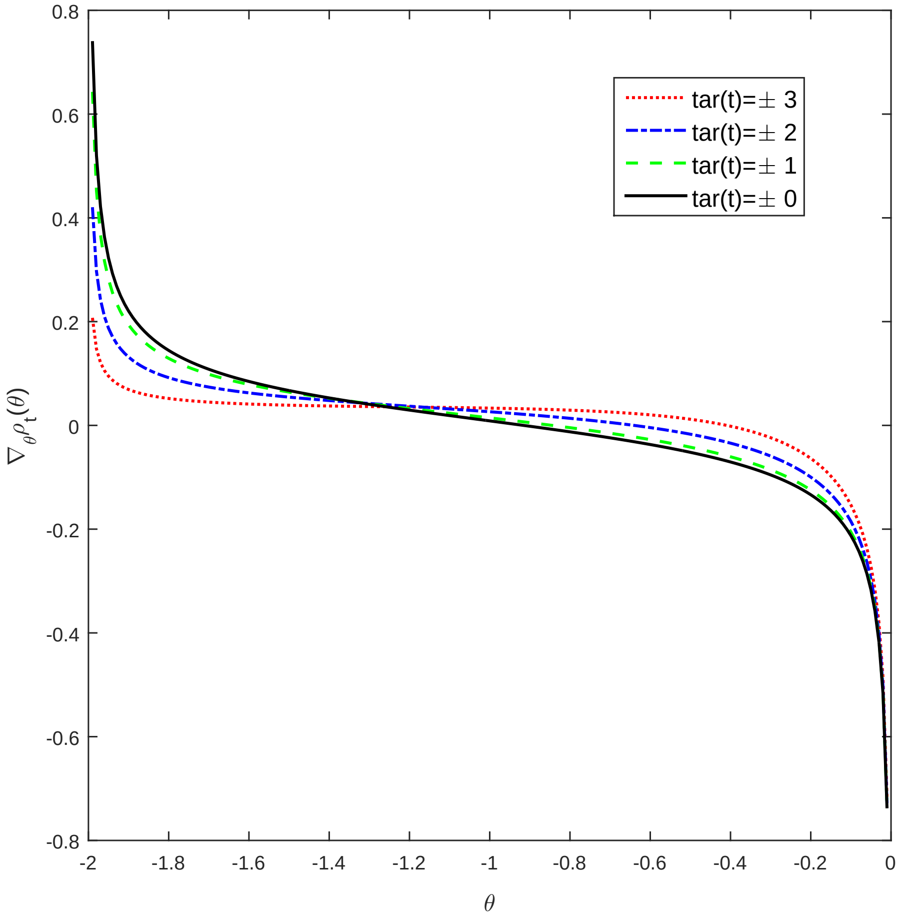


Figure 3.6: The derivative of $\rho_{r_t}(\theta)$ with respect to θ.

Chapter 4

Online MDPs with Policy Iteration

The *online Markov decision process* (MDP) is a generalization of the classical Markov decision process that incorporates changing reward functions. In this chapter, we propose practical online MDP algorithms with *policy iteration* algorithm by parameterizing the value function. We further theoretically establish a sublinear regret bound. A notable advantage of the proposed algorithm is that it can be easily combined with function approximation, and thus large and possibly continuous state spaces can be efficiently handled. Through experiments, we demonstrate the usefulness of the proposed algorithm.

4.1 Introduction

In this section, we present involved preliminaries of the online MDP problem. As we mentioned in previous chapters, the regret with respect to the best offline time independent policy is defined as:

$$L_{\mathcal{A}}(T) = R_{\pi^*}(T) - R_{\mathcal{A}}(T),$$

where $R_{\pi^*}(T)$ is the return of the best offline time independent policy π^*:

$$R_{\pi^*}(T) = \mathbb{E}_{\pi^*}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] = \sup_{\pi \in \Pi} \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right],$$

where $\mathbb{E}_{\pi}[\cdot]$ denotes the expectation over the state-action joint distribution given policy π. An important fact is that the regret we consider here is different from previous literature (Even-Dar et al., 2003, 2009; Zimin and Neu, 2013; Dick et al., 2014): we compare the performance of algorithm $\mathcal{A}$ against the best offline policy *within* a specific policy set Π. Namely instead of the best deterministic greedy policy, we consider a set of "efficient" policies, e.g., Gibbs policies with all possible parameters.

We expect that the regret $L_{\mathcal{A}}(T)$ is sublinear with respect to T, which means that the regret tends to zero as T tends to infinity and thus algorithm $\mathcal{A}$ performs as well as the best offline policy π^* asymptotically.

Next, we introduce some necessary notions for discussing online MDP problems. Recall the evaluation measures introduced in Chapter 1, there are different types of measures defined for different MDP problems. In this chapter, we define the value function by using the average evaluations as

$$\mathcal{V}_r^{\pi}(\boldsymbol{s}) = \mathbb{E}_{\pi}\left[\sum_{i=1}^{\infty}(r(\boldsymbol{s}_i, \boldsymbol{a}_i) - \rho_r(\pi))|\boldsymbol{s}_1 = \boldsymbol{s}\right],$$

For any arbitrary reward function $r(\boldsymbol{s}, \boldsymbol{a})$ and transition probability $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$, there exist at least one optimal policy $\pi^+ \in \Pi$ such that

$$\mathcal{V}_r^{\pi^+}(\boldsymbol{s}) \geq \mathcal{V}_r^{\pi}(\boldsymbol{s}), \forall \pi \in \Pi, \boldsymbol{s} \in S,$$

$$\rho_r(\pi^+) \geq \rho_r(\pi), \forall \pi \in \Pi.$$

Similarly, the state-action function is defined as

$$Q_r^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\pi}\left[\sum_{i=1}^{\infty}(r(\boldsymbol{s}_i, \boldsymbol{a}_i) - \rho_r(\pi))|\boldsymbol{s}_1 = \boldsymbol{s}, \boldsymbol{a}_1 = \boldsymbol{a}\right].$$

Since the optimal value function leads to the optimal policy, MDP is often solved by deriving the optimal value function (Sutton and Barto, 1998). So far, various efficient methods for approximating the optimal value function have been proposed. However, these algorithms were not proved to converge to the value function corresponding to the optimal deterministic policy. For this reason, in this paper we only consider the stochastic policy, since the convergence guarantee is provided (Tsitsiklis and Roy, 1999).

4.2 Online MDPs with Policy Iteration

In Chapter 3, we showed that the OPG algorithm achieved a sublinear regret under the concavity assumption by parameterizing the policy space. However, this concavity assumption is not always hold in some real problems. In this section, we introduce another proposed method for online MDPs which intends to parameterize the value function space. The key idea of the proposed algorithm is motivated by the Lazy FPL algorithm by Yu et al. (2009), which performs linear programming to obtain the 'leader' policy. As Yu et al. (2009) pointed out, solving linear programming may not be appropriate for problems with large (continuous) state space. For this reason, we employ a policy iteration type method together with a stochastic policy in our proposed method.

4.2.1 Proposed Algorithm

Firstly, we define the policy improvement operator $\Gamma : \pi(\boldsymbol{a}|\boldsymbol{s}) = \Gamma(r(\boldsymbol{s},\boldsymbol{a}), V(\boldsymbol{s}))$, where $r(\boldsymbol{s},\boldsymbol{a})$ is an arbitrary reward function, $V(\boldsymbol{s})$ is an arbitrary value function. Below we use $\Gamma(r, V)$ instead of $\Gamma(r(\boldsymbol{s},\boldsymbol{a}), V(\boldsymbol{s}))$ for notational simplicity. Now we introduce two assumptions on the defined operator Γ.

Assumption 8. *For an arbitrary reward function r and two arbitrary value functions $V_1(\boldsymbol{s})$ and $V_2(\boldsymbol{s})$, the policies $\pi_1 = \Gamma(r, V_1)$ and $\pi_2 = \Gamma(r, V_2)$ satisfy*

$$\|\pi_1(\boldsymbol{s},\cdot) - \pi_2(\boldsymbol{s},\cdot)\|_1 \leq \xi\|V_1(\cdot) - V_2(\cdot)\|_\infty,$$

where $\xi > 0$ is the Lipschitz constant depending on the specific policy model. $\|\cdot\|_1$ denotes the L_1 norm, $\|\cdot\|_\infty$ denotes the infinity norm in this chapter.

Assumption 9. *For an arbitrary value function $V(\boldsymbol{s})$ and two arbitrary reward functions $r(\boldsymbol{s},\boldsymbol{a})$ and $r'(\boldsymbol{s},\boldsymbol{a})$, the policies $\pi = \Gamma(r, V)$ and $\pi' = \Gamma(r', V)$ satisfy*

$$\|\pi(\boldsymbol{s},\cdot) - \pi'(\boldsymbol{s},\cdot)\|_1 \leq \xi\|r(\boldsymbol{s},\cdot) - r'(\boldsymbol{s},\cdot)\|_\infty,$$

The *Gibbs policy* is a popular model which was demonstrated to work well:

$$\pi(\boldsymbol{a}|\boldsymbol{s}) = \frac{\exp\frac{1}{\kappa}\left(r(\boldsymbol{s},\boldsymbol{a}) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})V(s')\right))}{\sum_{\boldsymbol{a}'\in A}\exp\frac{1}{\kappa}\left(r(\boldsymbol{s},\boldsymbol{a}') + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}')V(s')\right)},$$

where κ is the exploration parameter. We can show that the Gibbs policy satisfies Assumption 8 and Assumption 9 (the proofs are provided in Appendix 4.6.1).

Throughout this chapter, we only consider stochastic policies that satisfy the above two assumptions. Let Π be the set of policies generated by the operator Γ. Then our proposed *online MDP with policy iteration (OMDP-PI) algorithm* is given as follows:

- Initialize the value function $V_0(\boldsymbol{s}) = 0, \forall \boldsymbol{s} \in S$.
- for $t = 1, \ldots, \infty$
  1. Observe the current state $\boldsymbol{s}_t = \boldsymbol{s}$.
  2. Improve the policy as $\pi_t = \Gamma(\hat{r}_{t-1}, V_{t-1})$, where
  $$\hat{r}_{t-1}(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{t-1} \sum_{k=1}^{t-1} r_k(\boldsymbol{s}, \boldsymbol{a}).$$
  3. Take action $\boldsymbol{a}_t = \boldsymbol{a}$ by following π_t.
  4. The reward function $r_t(\boldsymbol{s}, \boldsymbol{a})$ is revealed.
  5. Update the value function according to
  $$V_t(\boldsymbol{s}) = (1 - \gamma_t) V_{t-1}(\boldsymbol{s}) + \gamma_t \mathcal{V}_{r_t}^{\pi_t}(\boldsymbol{s}), \tag{4.1}$$
  where the step size is $\gamma_t = 1/t$.

It is well known (Sutton and Barto, 1998) that the value function satisfies

$$\mathcal{V}_r^{\pi}(\boldsymbol{s}) = \mathbb{E}_{\pi} \left[r(\boldsymbol{s}, \boldsymbol{a}) - \rho_r(\pi) + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) \mathcal{V}_r^{\pi}(\boldsymbol{s}') \right].$$

The above equation can be rewritten in matrix form as

$$\mathcal{V}_r^{\pi} = R(\pi) - \boldsymbol{e}_{|S|} \rho_r(\pi) + P^{\pi} \mathcal{V}_r^{\pi}, \tag{4.2}$$

where $\mathcal{V}_r^{\pi}$ is the $|S|$-dimensional column vector whose $\boldsymbol{s}$th element is $\mathcal{V}_r^{\pi}(\boldsymbol{s})$. $R(\pi)$ is the $|S|$-dimensional column vector whose $\boldsymbol{s}$th element is $\sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|\boldsymbol{s}) r(\boldsymbol{s}, \boldsymbol{a})$. P^{π} is the transition matrix induced by the policy π, whose $\boldsymbol{s}\boldsymbol{s}'$th element is $p^{\pi}(\boldsymbol{s}|\boldsymbol{s}') =$

$\sum_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s})p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$. $\boldsymbol{e}_{|S|}$ is the $|S|$-dimensional column vector with all ones. It is well known (Sutton and Barto, 1998) that the above equation has no unique solution. Here we introduce the following constraint on the value function:

$$\mathbb{E}_{\boldsymbol{s}\sim d_\pi(\boldsymbol{s})}[\mathcal{V}_r^\pi(\boldsymbol{s})] = \mathbb{E}_{\boldsymbol{s}\sim d_\pi(\boldsymbol{s}),\boldsymbol{a}\sim\pi}\left[\sum_{i=1}^{\infty}(r(\boldsymbol{s},\boldsymbol{a}) - \rho_r(\pi))\right] = 0.$$

By this constraint, the solution of Equ.(4.2) becomes unique and satisfies

$$\mathcal{V}_r^\pi = R(\pi) - \boldsymbol{e}_{|S|}\rho_r(\pi) + P^\pi\mathcal{V}_r^\pi - \boldsymbol{e}_{|S|}d_\pi^\top\mathcal{V}_r^\pi, \tag{4.3}$$

where d_π is the $|S|$-dimensional column vector whose $\boldsymbol{s}$th element is $d_\pi(\boldsymbol{s})$.

Then the update rule (4.1) can be expressed in closed form as

$$V_t = (1-\gamma_t)V_{t-1} + \gamma_t(\boldsymbol{I}_{|S|} - P^{\pi_t} + \boldsymbol{e}_{|S|}d_{\pi_t}^\top)^{-1}(R_t(\pi_t) - \boldsymbol{e}_{|S|}\rho_{r_t}(\pi_t)).$$

Since the stationary distribution can be obtain by the eigenvector corresponding to the unit eigenvalue, we can calculate $\rho_{r_t}(\pi_t)$ directly. Then, $V_t(\boldsymbol{s})$ can be obtained directly without solving an optimization problem when the state space is not large (continuous). In the following sections, we will introduce an approximation method to handle large (continuous) state space problems.

4.2.2 Regret Analysis

In this section, we provide a regret analysis for the proposed OMDP-PI algorithm. Firstly, we introduce several essential assumptions involved in the proof. Similarly to the previous works (Even-Dar et al., 2003, 2009; Yu et al., 2009; Neu et al., 2010b, 2014; Ma et al., 2014), we assume the following conditions.

Assumption 10. *For all $\pi \in \Pi$, there exist a positive constant τ such that two arbitrary state distributions $d(\boldsymbol{s})$ and $d'(\boldsymbol{s})$ satisfy*

$$\sum_{\boldsymbol{s}\in S}\sum_{\boldsymbol{s}'\in S}|d(\boldsymbol{s}) - d'(\boldsymbol{s})|p^\pi(\boldsymbol{s}'|\boldsymbol{s}) \le e^{-1/\tau}\sum_{\boldsymbol{s}\in S}|d(\boldsymbol{s}) - d'(\boldsymbol{s})|.$$

Assumption 11. *The reward functions satisfy*

$$r_t(\boldsymbol{s},\boldsymbol{a}) \in [0,1], \forall\boldsymbol{s} \in S, \forall\boldsymbol{a} \in A, \forall t = 1,\ldots,T.$$

Under these assumptions, the regret of the OMDP-PI algorithm for a policy set Π is bounded as follows:

Theorem 4.1. *After T time steps, the regret against the best offline time independent policy of the OMDP-PI algorithm is bounded as*

$$L_{\mathrm{OMDP-PI}}(T) \leq \frac{2-e^{-1/\tau}}{1-e^{-1/\tau}} C\xi T^{C_v} + \left(\frac{6\tau\xi(2-e^{-1/\tau})}{1-e^{-1/\tau}} + 2\tau^3\right)\ln T$$
$$+ \left(\frac{6\tau\xi(2-e^{-1/\tau})}{1-e^{-1/\tau}} + 2\tau^3 + 2\tau^3 e^{\tau+2} + 4\tau\right),$$

where $C = 6\tau(2 - C_v + \frac{1}{C_v} + \frac{1-C_v}{1+C_v})$, $C_v = \xi C_\pi$, and C_π is a positive constant such that for all $\pi_1, \pi_2 \in \Pi$,

$$\|\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}\|_\infty \leq C_\pi \|\pi_1 - \pi_2\|_1.$$

The existence of C_π is proved in Appendix 4.6.5.

Remark 4.2. *The regret bound in Theorem 4.1 is sublinear with respect to T when $C_v < 1$. However, the quality of the policy is limited when C_v is small. Since the smaller the constant C_v is, the poorer the performance of the best offline policy is. In an extreme case, where all the policies in the set Π perform equally, when $C_v = 0$.*

To prove the claimed result in Theorem 4.1, we decompose the regret into three parts in the same way as previous works (Even-Dar et al., 2003, 2009; Abbasi-Yadkori et al., 2013; Ma et al., 2014):

$$L_{\mathcal{A}}(T) = \left(\mathbb{E}_{\pi^*}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] - \sum_{t=1}^{T}\rho_{r_t}(\pi^*)\right) + \left(\sum_{t=1}^{T}\rho_{r_t}(\pi^*) - \sum_{t=1}^{T}\rho_{r_t}(\pi_t)\right)$$
$$+ \left(\sum_{t=1}^{T}\rho_{r_t}(\pi_t) - \mathbb{E}_{\pi_t}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]\right).$$

The first term has been analyzed in previous works (Even-Dar et al., 2003, 2009; Ma et al., 2014), which is bounded as

$$\mathbb{E}_{\pi^*}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] - \sum_{t=1}^{T}\rho_{r_t}(\pi^*) \leq 2\tau.$$

Below, we bound the second and the third terms in Lemma 4.3 and Lemma 4.4 which are proved in Appendix 4.6.2 and Appendix 4.6.3.

Lemma 4.3. *After T time steps, the policy sequence $\pi_1, \ldots, \pi_T$ given by OMDP-PI and the best offline policy $\pi^* \in \Pi$ satisfy*

$$\sum_{t=1}^{T} \rho_{r_t}(\pi^*) - \sum_{t=1}^{T} \rho_{r_t}(\pi_t) \le \frac{2 - e^{-1/\tau}}{1 - e^{-1/\tau}} \left(C\xi T^{C_v} + 6\tau\xi \ln T + 6\tau\xi \right),$$

where $C = 6\tau(2 - C_v + \frac{1}{C_v} + \frac{1-C_v}{1+C_v})$.

Lemma 4.4. *After T time steps, the policy sequence $\pi_1, \ldots, \pi_T$ given by OMDP-PI satisfies*

$$\sum_{t=1}^{T} \rho_{r_t}(\pi_t) - \mathbb{E}_{\pi_t} \left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] \le 2\tau^3 \ln T + 2\tau^3 + 2\tau^3 e^{(\tau+2)} + 2\tau.$$

Summarizing these bounds, we can obtain Theorem 4.1.

4.2.3 OMDP-PI Algorithm with Approximation

When considering large (continuous) state space in online MDP problems, it is essential to apply a function approximation technique. Tsitsiklis and Roy (1999) introduced the linear function approximation of the value function for stochastic policies. A significant benefit of the linear approximation is that the convergence guarantee is provided (Tsitsiklis and Roy, 1999). Below we present their theoretical results for discrete (possibly continuous) state space.

By following the same idea as Tsitsiklis and Roy (1999), we use the linear approximation of the value function:

$$\hat{\mathcal{V}}(\boldsymbol{s}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(s),$$

where $\boldsymbol{\theta} \in \Theta$ is the approximation parameter, and $\Theta \subset \mathbb{R}^K$ is the parameter space, $\boldsymbol{\phi}(\boldsymbol{s})$ is the basis function. At each time step t, the value function $\mathcal{V}_{r_t}^{\pi_t}(\boldsymbol{s})$ is approximated as follows:

- for $i = 1, 2, \ldots$ until convergence

  1. Observe the state $\boldsymbol{s}_i$.
  2. Take action $\boldsymbol{a}_i$ following π_t.

3. Observe the next state $\boldsymbol{s}_{i+1}$ and the reward $r_t(\boldsymbol{s}_i, \boldsymbol{a}_i)$
4. Update the approximation parameter as

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha_t(r_t(\boldsymbol{s}_i, \boldsymbol{a}_i) - \hat{\rho}^{\pi_t}_{r_t}(i) + \boldsymbol{\theta}_i^\top \boldsymbol{\phi}(\boldsymbol{s}_{i+1}) - \boldsymbol{\theta}_i^\top \boldsymbol{\phi}(\boldsymbol{s}_i))$$

and

$$\hat{\rho}^{\pi_t}_{r_t}(i+1) = (1-\alpha_t)\hat{\rho}^{\pi_t}_{r_t}(i) + \alpha_t r_t(\boldsymbol{s}_i, \boldsymbol{a}_i),$$

where the step size α_t satisfies

$$\sum_{t=1}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty.$$

The approximation parameter was proved to converge to the unique solution of the following equation (Tsitsiklis and Roy, 1999):

$$\mathcal{P}(R_t(\pi_t) - \boldsymbol{e}_{|S|}\rho_{r_t}(\pi_t) + P^{\pi_t}\boldsymbol{\theta}^\top \boldsymbol{\phi}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}, \tag{4.4}$$

where $R_t(\pi_t)$ is the $|S|$-dimensional column vector whose $\boldsymbol{s}$th element is $r_t(\boldsymbol{s}, \pi_t) = \sum_{\boldsymbol{a} \in A} \pi_t(\boldsymbol{a}|\boldsymbol{s}) r_t(\boldsymbol{s}, \boldsymbol{a})$. $\mathcal{P}$ is the projection operator such that for all $V \in \mathbb{R}^{|S|}$,

$$\mathcal{P}(V) = \operatorname{argmin}_{\bar{V} \in \{\boldsymbol{\theta}^\top \boldsymbol{\phi} | \boldsymbol{\theta} \in \mathbb{R}^K\}} \|V - \bar{V}\|_{D^{\pi_t}},$$

where D^{π_t} is the diagonal matrix with the stationary distribution on the diagonal. It is clear that $\mathcal{P}$ is the projection from the $|S|$-dimensional real space to the space spanned by the basis function. The approximation sequence $\hat{\rho}^{\pi_t}_{r_t}(1), \hat{\rho}^{\pi_t}_{r_t}(2), \ldots$ satisfies

$$\lim_{i \to \infty} \hat{\rho}^{\pi_t}_{r_t}(i) \to \rho_{r_t}(\pi_t), \text{ with probability } 1.$$

Furthermore, by using Theorem 3 in Tsitsiklis and Roy (1999), the approximation error is bounded as

$$\|(\boldsymbol{I}_{|S|} - \boldsymbol{e}_{|S|} d_{\pi_t}^\top)\boldsymbol{\theta}_t^{*\top} \boldsymbol{\phi} - \mathcal{V}^{\pi_t}_{r_t}\|_{D_{\pi_t}} \leq \frac{1}{\sqrt{1 - e^{-2/\tau}}} \inf_{\boldsymbol{\theta} \in \mathbb{R}^K} \|(\boldsymbol{I}_{|S|} - \boldsymbol{e}_{|S|} d_{\pi_t}^\top)\boldsymbol{\theta}^\top \boldsymbol{\phi} - \mathcal{V}^{\pi_t}_{r_t}\|_{D_{\pi_t}},$$

where $\boldsymbol{\theta}_t^*$ is the unique solution to Eq.(4.4) at time step t. We observe that the approximation error is zero when the linear approximation model is capable of exactly recovering the true value function.

4.3 Online MDPs with Stochastic Iteration

In this section, we introduce a more general framework of our proposed method for online MDPs. More specifically, we extend our algorithm to use *stochastic iteration* (Bertsekas and Tsitsiklis, 1996) for policy evaluation together with policy improvement to solve online MDPs.

A general form of the stochastic iteration algorithm (Szita et al., 2002; Csáji and Monostori, 2008) can be expressed as

$$V_t(\boldsymbol{s}) = (1 - \gamma_t(\boldsymbol{s}))V_{t-1}(\boldsymbol{s}) + \gamma_t(\boldsymbol{s}) \left((H_t V_{t-1})(\boldsymbol{s}) + w_t(\boldsymbol{s})\right), \tag{4.5}$$

where $V_t \in \mathbb{R}^{|S|}$, $H_t : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}, \forall t = 1, \dots, T$ is an operator on value functions, γ_t is the step size, and $w_t(\boldsymbol{s})$ is a noise term. Similarly to the Eq.(4.5), we define the update rule as

$$V_t(\boldsymbol{s}) = (1 - \gamma_t(\boldsymbol{s}))V_{t-1}(\boldsymbol{s}) + \gamma_t(\boldsymbol{s})((H_t^{\pi_t} V_{t-1})(\boldsymbol{s}) + w_t(\boldsymbol{s})), \tag{4.6}$$

where $\pi_t = \Gamma(\hat{r}_{t-1}, V_{t-1})$ satisfies Assumption 8 and Assumption 9. Note that the update rule (4.6) is different from standard stochastic iteration (4.5), where the operator H_t is replaced by the controlled operator H_t^{π} which the OMDP-PI algorithm uses: $H_t^{\pi_t} V_{t-1}(\boldsymbol{s}) = \mathcal{V}_{r_t}^{\pi_t}(\boldsymbol{s})$. Additionally, we require the following assumptions.

Assumption 12. *The controlled operator H_t^{π} is a contraction mapping with respect to the value function. This means that, for two arbitrary value functions V and V' and two policies $\pi = \Gamma(r, V)$, $\pi' = \Gamma(r, V')$, there exist a no negative constant $\beta_t < 1$ such that*

$$\|H_t^{\pi} V - H_t^{\pi} V'\| \leq \beta_t \|V - V'\|,$$

and there exist a fixed function V_t^* satisfies

$$H_t V_t^* = V_t^*.$$

Assumption 13. *For all $t = 1, \dots, T$, the noisy terms $w_t(\boldsymbol{s})$ satisfy*

$$\mathbb{E}[w_t(\boldsymbol{s})] = 0 \text{ and } \mathbb{E}[w_t^2(\boldsymbol{s})] < C_w < \infty,$$

where C_w is a positive constant.

Assumption 14. *The step size γ_t satisfies*

$$\sum_{t=1}^{\infty} \gamma_t = \infty \text{ and } \sum_{t=1}^{\infty} \gamma_t^2 < \infty.$$

Assumption 15. *The value functions sequence $V_1(\boldsymbol{s}), \ldots, V_T(\boldsymbol{s})$ generated by Eq.*(4.6) *satisfies*

$$\lim_{T\to\infty} \|V_T^* - \max_{\pi\in\Pi} \mathcal{V}_{\hat{r}_T}^{\pi}\|_\infty = 0.$$

Then we have the following theorem:

Theorem 4.5. *If Assumptions 5-8 hold, the value function sequence $V_1(\boldsymbol{s}), \ldots, V_T(\boldsymbol{s})$ generated by Eq.*(4.6) *satisfies*

$$\lim_{T\to\infty} L_{\mathcal{A}}(T) = 0.$$

Proof. By using Theorem 20 in Csáji and Monostori (2008), we have

$$\lim_{T\to\infty} \|V_T - \mathcal{V}_{\hat{r}_T}^{\pi^*}\|_\infty = 0,$$

where $\pi^* = \operatorname{argmax}_{\pi\in\Pi} \rho_{\hat{r}_T}(\pi)$ is the best offline policy. Since $\pi_{T+1} = \Gamma(\hat{r}_T, V_T)$ and π_T, we obtain

$$\begin{aligned}
\lim_{T\to\infty} \|\pi_T - \pi^*\|_1 &\leq \lim_{T\to\infty} (\|\pi_{T+1} - \pi^*\|_1 + \|\pi_{T+1} - \pi_T\|_1) \\
&\leq \lim_{T\to\infty} (\xi\|V_T - \mathcal{V}_{\hat{r}_T}^{\pi^*}\|_\infty + \|\pi_{T+1} - \pi_T\|_1) \\
&\leq \lim_{T\to\infty} (\xi\|V_T - \mathcal{V}_{\hat{r}_T}^{pi^*}\|_\infty + \xi\|\hat{r}_{T+1} - \hat{r}_T\|_\infty + \xi\|V_T - V_{T-1}\|_\infty) \\
&= 0.
\end{aligned}$$

In the above derivation we used $\lim_{T\to\infty} \|V_T - V_{T-1}\|_\infty = 0$, which can be obtained by the update rule (4.6). The above result shows that the policies generated by the value sequence converges to the best offline policy as T goes to infinity. Hence, the claimed result hold by following the same line as the proof of Theorem 4.1. □

Many popular reinforcement learning algorithms based on value functions such as the *temporal difference (TD) learning* algorithm (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998; Sutton, 1988) and the *SARSA* algorithm (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998) can be regarded as stochastic iteration. Theorem 4.5 shows that any stochastic iteration method that satisfies Assumptions 5-8 could be used to derive an online MDPs algorithm with sublinear regret.

4.4 Experiments

In this section, we experimentally illustrate the behavior of the proposed online algorithm.

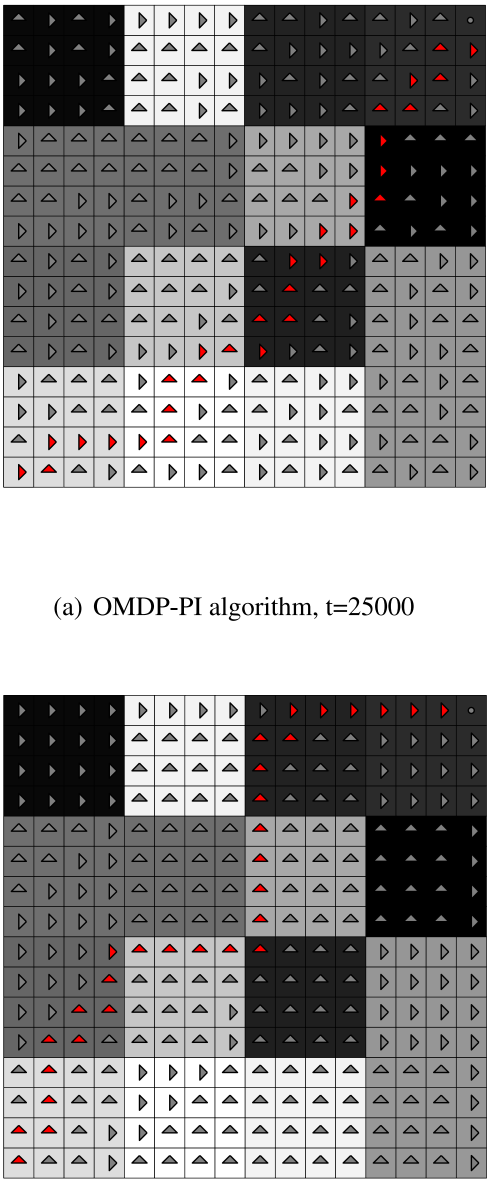
The goal of the *grid world* problem is to let an agent walk in the grid environment from the start block to the destination block. We conduct experiments on the grid world based on the *Inverse Reinforcement Learning* (IRL) toolkit[1](Levine et al., 2011).

First of all, we construct a typical grid world environment with 16×16 states and 2 actions in each state, which correspond to moving east and north. Each action has a 30% chance of moving in the other direction. The 256 states are further joined into 16 super-grids, each of which consists of 4×4 states with the same reward.

In each episode, the agent tries to find a trajectory from the south-west corner to the north-east corner, with the highest cumulative rewards. In the north or east border states, the agent can only move east or north. Different from the standard grid world problem, we set $T = 100000$ and randomly change the rewards at episodes $t = 1, 5001, 10001, ..., 95001$. The proposed algorithm and the best offline algorithm (obtained using the standard MDP solver included in the IRL toolkit) are run on the grid world.

We show the trajectories found by the offline policy and the proposed OMDP-PI algorithm at episodes $t = 25000, 50000, 75000, 100000$ in Figure 4.1, Figure 4.2, Figure 4.3, and Figure 4.4, respectively. The darker the state is, the lower average reward it has. The direction of triangles shows the obtained policies. The states with red triangles indicate trajectories of the agent. Figure 4.5 shows the average regret and cumulative reward as functions of the number of episodes.

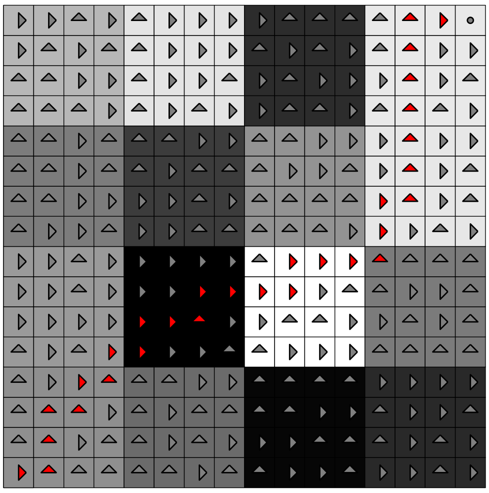
[1] http://graphics.stanford.edu/projects/gpirl



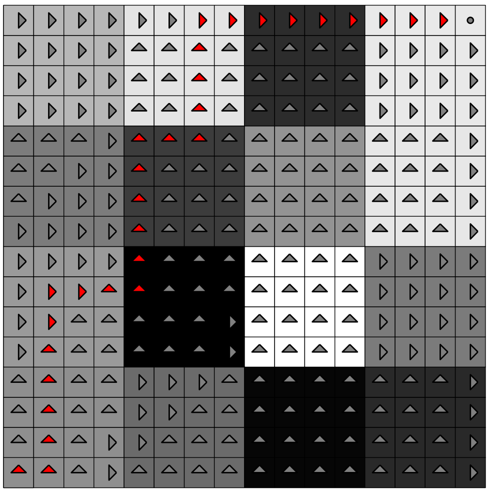
(a) OMDP-PI algorithm, t=25000

(b) Best offline policy, t=25000

Figure 4.1: Experiments on grid worlds (t=25000).

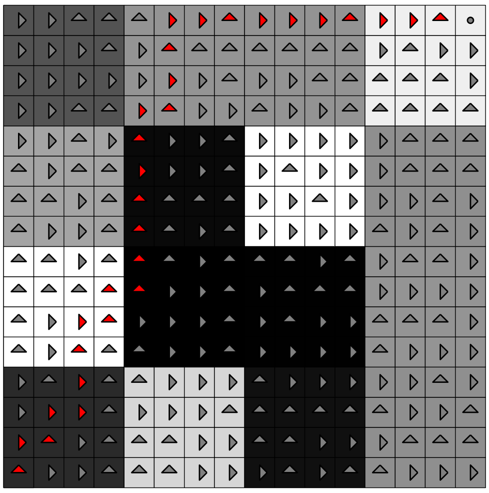


(a) OMDP-PI algorithm, t=50000

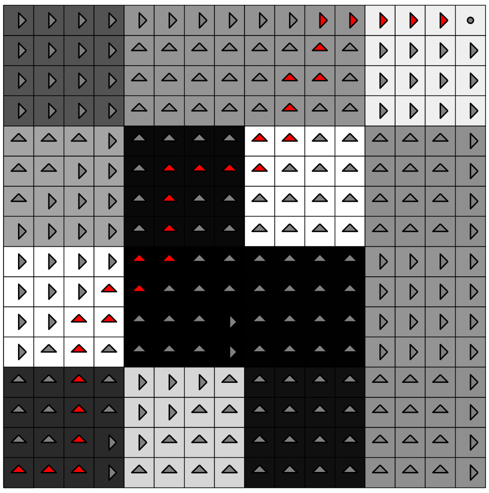


(b) Best offline policy, t=50000

Figure 4.2: Experiments on grid worlds (t=50000).

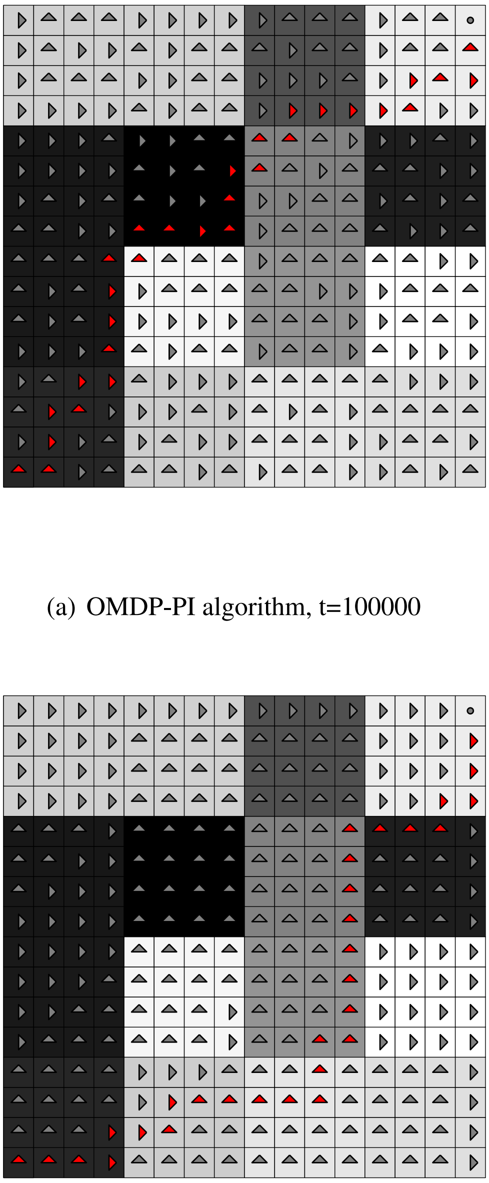


(a) OMDP-PI algorithm, t=75000



(b) Best offline policy, t=75000

Figure 4.3: Experiments on grid worlds (t=75000).



(a) OMDP-PI algorithm, t=100000

(b) Best offline policy, t=100000

Figure 4.4: Experiments on grid worlds (t=100000).

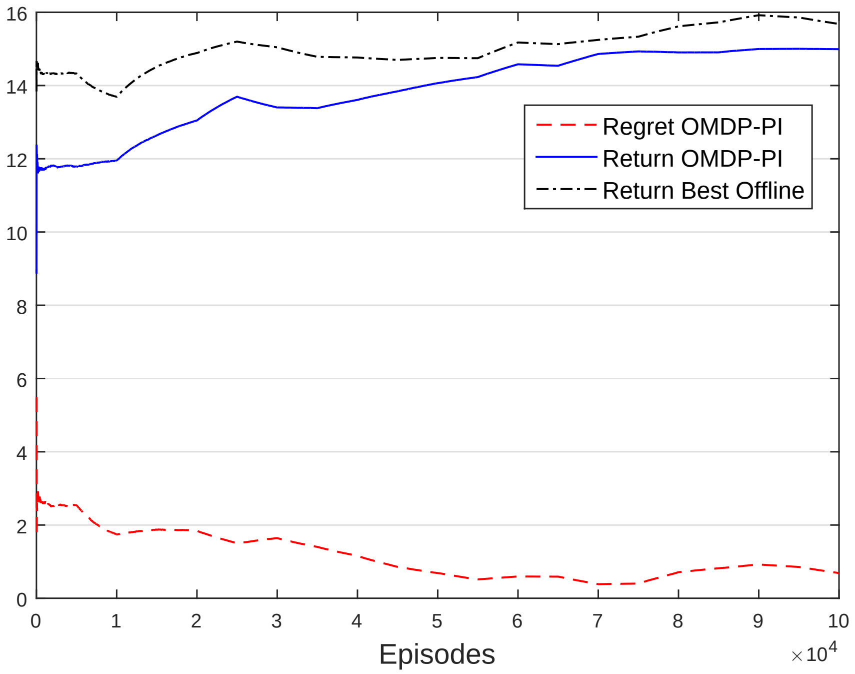


Figure 4.5: Regrets and rewards.

The results in Figure 4.5 show that the regret of the OMDP-PI algorithms vanishes, substantiating that our theoretical analysis is valid.

4.5 Comparison with Previous Work

In this section, we compare the proposed OMDP-PI algorithm with previous work.

- Expert algorithm based methods (Even-Dar et al., 2003, 2009; Neu et al., 2010a, 2014): The basic idea of expert algorithm based methods is to put an expert algorithm in every state. By taking a close look at these algorithms, the idea does not take advantage of the state structure of the MDP problem. The OMDP-PI algorithm can be easily combined with function approximation. Since it is popular to simplify the large state space problem by using the linear span of the state features, the OMDP-PI algorithm is natural to handle the large state space online MDPs.

- Online linear optimization based methods (Zimin and Neu, 2013; Dick et al., 2014): By introducing the stationary occupancy measures over state-action pairs, the online MDP problems can be solved as the online linear optimization problems. The $O(\sqrt{T})$ regret bounds are proved for fixed time horizon online MDPs. More specifically, the step size parameter is optimized by using the length of the time horizon T. Moreover, the stationary occupancy measures are defined over finite state and action spaces, and thus it is not clear that whether the state-action probability density function could be learned by using their propose methods without parametrization. The OMDP-PI algorithm with function approximation parameterized the state-action density through the linear model of the value function.

- Linear programming based method (Yu et al., 2009): Our OMDP-PI is motivated by the Lazy-FPL algorithm, which solves a linear programming problem at the end of each phase. Instead of obtaining the best policy by the linear programming, the OMDP-PI algorithm obtains the value function of the current policy which is much more efficient than the linear programming. As we showed in the update rule, the policy evaluation could be performed in $O(|S|^{2.3728639} + |S|^2|A|)$ where the matrix inversion could be solved in $O(|S|^{2.3728639})$ (Le Gall, 2014).

4.6 Proofs of Theorems

In this section, we present the proofs of all the theorems involved in this chapter.

4.6.1 Proof of Gibbs policy

Proof. Define $\nu(\boldsymbol{s}, \boldsymbol{a}', \boldsymbol{s}') = r(\boldsymbol{s}, \boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}')V(\boldsymbol{s}')$ and $\Delta\nu(\boldsymbol{s}, \boldsymbol{a}', \boldsymbol{s}') = \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}')(V'(\boldsymbol{s}') - V(\boldsymbol{s}'))$, the KL divergence of two Gibbs policies π, π'

generated by two different value function V and V' is

$$\begin{aligned}
&D(\pi(\cdot|\boldsymbol{s}) \parallel \pi'(\cdot|\boldsymbol{s})) \\
&= \mathbb{E}_\pi \left[\sum_{\boldsymbol{s}' \in S} \frac{1}{\kappa} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V(\boldsymbol{s}') - V'(\boldsymbol{s}')\right) \right] \\
&\quad + \log \frac{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} \left(r(\boldsymbol{s},\boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}') V'(\boldsymbol{s}') \right)}{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} \left(r(\boldsymbol{s},\boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}') V(\boldsymbol{s}') \right)} \\
&= \mathbb{E}_\pi \left[\frac{1}{\kappa} \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V(\boldsymbol{s}') - V'(\boldsymbol{s}')\right) \right] \\
&\quad + \log \frac{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} (\nu(\boldsymbol{s},\boldsymbol{a}',\boldsymbol{s}') + \Delta\nu(\boldsymbol{s},\boldsymbol{a}',\boldsymbol{s}'))}{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} (r(\boldsymbol{s},\boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}') V(\boldsymbol{s}'))} \\
&= \mathbb{E}_\pi \left[\frac{1}{\kappa} \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V(\boldsymbol{s}') - V'(\boldsymbol{s}')\right) \right] \\
&\quad + \log \frac{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} \left(\nu(\boldsymbol{s},\boldsymbol{a}',\boldsymbol{s}') \right) \exp \frac{1}{\kappa} \Delta\nu(\boldsymbol{s},\boldsymbol{a}',\boldsymbol{s}')}{\sum_{\boldsymbol{a}' \in A} \exp \frac{1}{\kappa} \left(r(\boldsymbol{s},\boldsymbol{a}') + \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}') V(\boldsymbol{s}') \right)} \\
&= \mathbb{E}_\pi \left[\frac{1}{\kappa} \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V(\boldsymbol{s}') - V'(\boldsymbol{s}')\right) \right] \\
&\quad + \log \mathbb{E}_\pi \left[\frac{1}{\kappa} \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V'(\boldsymbol{s}') - V(\boldsymbol{s}')\right) \right] \\
&\leq \frac{\| \frac{1}{\kappa} \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \left(V(\boldsymbol{s}') - V'(\boldsymbol{s}')\right) \|_\infty^2}{4}.
\end{aligned}$$

From the Pinsker's inequality, there is

$$\|\pi(\cdot|\boldsymbol{s}) - \pi'(\cdot|\boldsymbol{s})\|_1 \leq \frac{\|V(\boldsymbol{s}) - V'(\boldsymbol{s})\|_\infty}{\sqrt{2}\kappa},$$

Similarly, the KL divergence of two Gibbs policies π, π' generated by two different reward function $r(\boldsymbol{s},\boldsymbol{a})$ and $r'(\boldsymbol{s},\boldsymbol{a})$ is

$$\begin{aligned}
&D(\pi(\cdot|\boldsymbol{s})||\pi'(\cdot|\boldsymbol{s}))\\
&=\mathbb{E}_{\pi}\left[\frac{1}{k}(r(\boldsymbol{s},\cdot)-r'(\boldsymbol{s},\cdot))\right]\\
&\quad+\log\frac{\sum_{\boldsymbol{a}'\in A}\exp\frac{1}{\kappa}(r'(\boldsymbol{s},\boldsymbol{a}')+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}')V(\boldsymbol{s}'))}{\sum_{\boldsymbol{a}'\in A}\exp\frac{1}{\kappa}(r(\boldsymbol{s},\boldsymbol{a}')+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}')V(\boldsymbol{s}'))}\\
&=\mathbb{E}_{\pi}\left[\frac{1}{k}(r(\boldsymbol{s},\cdot)-r'(\boldsymbol{s},\cdot))\right]\\
&\quad+\log\frac{\sum_{\boldsymbol{a}'\in A}\exp\frac{1}{\kappa}(r(\boldsymbol{s},\boldsymbol{a}')+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}')V(\boldsymbol{s}')+r'(\boldsymbol{s},\boldsymbol{a}')-r(\boldsymbol{s},\boldsymbol{a}'))}{\sum_{\boldsymbol{a}'\in A}\exp\frac{1}{\kappa}(r(\boldsymbol{s},\boldsymbol{a}')+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}')V(\boldsymbol{s}'))}\\
&=\mathbb{E}_{\pi}\left[\frac{1}{\kappa}(r(\boldsymbol{s},\cdot)-r'(\boldsymbol{s},\cdot))\right]\\
&\quad+\log\mathbb{E}_{\pi}\left[\frac{1}{\kappa}(r'(\boldsymbol{s},\cdot)-r(\boldsymbol{s},\cdot))\right]\\
&\leq\frac{\|\frac{1}{\kappa}(r(\boldsymbol{s},\cdot)-r'(\boldsymbol{s},\cdot))\|_{\infty}^{2}}{4}.
\end{aligned}$$

From the Pinsker's inequality, we can conclude the proof as

$$\|\pi(\cdot|\boldsymbol{s})-\pi'(\cdot|\boldsymbol{s})\|_1\leq\frac{\|r(\boldsymbol{s},\cdot)-r'(\boldsymbol{s},\cdot)\|_{\infty}}{\sqrt{2}\kappa}.$$

which concludes the proof. □

4.6.2 Proof of Lemma 4.3

Proposition 4.6. *The value functions sequence $V_1(\boldsymbol{s}),\dots,V_T(\boldsymbol{s})$ generated by the Equ*(4.1) *satisfies*

$$\|\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\cdot)-V_t(\cdot)\|_{\infty}\leq CC_v(t+1)^{C_v-1},$$

where $C=6\tau(2-C_v+\frac{1}{C_v}+\frac{1-C_v}{1+C_v})$*, and* $\pi_t^*=\operatorname{argmax}_{\pi\in\Pi}\rho_{\hat{r}_t}(\pi)$.

By Proposition 3 in Ma et al. (2014) and Proposition 4.1 in Yu et al. (2009), we obtain the following result

$$\sum_{t=1}^{T}(\rho_{r_t}(\pi^*)-\rho_{r_t}(\pi_t))\leq\sum_{t=1}^{T}(\rho_{r_t}(\pi_t^*)-\rho_{r_t}(\pi_t))\leq\sum_{t=1}^{T}\frac{2-e^{-1/\tau}}{1-e^{-1/\tau}}\|\pi_t^*-\pi_t\|_1$$

The result in Proposition 4.6 leads the following inequalities

$$
\begin{aligned}
&\sum_{t=1}^{T}\left(\rho_{r_t}(\pi^*) - \rho_{r_t}(\pi_t)\right) \\
&\le \sum_{t=1}^{T}\frac{2-e^{-1/\tau}}{1-e^{-1/\tau}}(\|\pi^*_{t-1}-\pi_t\|_1 + \|\pi^*_t - \pi^*_{t-1}\|_1) \\
&\le \sum_{t=1}^{T}\frac{2-e^{-1/\tau}}{1-e^{-1/\tau}}\xi(\|\mathcal{V}_{\hat{r}_{t-1}}^{\pi^*_{t-1}} - V_{t-1}\|_\infty + \frac{4\tau+2}{t}) \\
&\le \frac{2-e^{-1/\tau}}{1-e^{-1/\tau}}\left(\frac{C\xi}{C_v}T^{C_v} + 6\tau\xi\ln T + 6\tau\xi\right).
\end{aligned}
$$

4.6.3 Proof of Lemma 4.4

The proof is following the same line as previous works(Even-Dar et al., 2003, 2009; Ma et al., 2014), we rewrite the proof with our notations. By the definition of the expected average reward function, we have

$$
\begin{aligned}
&\sum_{t=1}^{T}\rho_{r_t}(\pi_t) - \mathbb{E}_{\pi_t}\left[\sum_{t-1}^{T} r_t(\boldsymbol{s}_t,\boldsymbol{a}_t)\right] \\
&= \sum_{t=1}^{T}\sum_{\boldsymbol{s}\in S}\sum_{\boldsymbol{a}\in A}\left(d_{\pi_t}(\boldsymbol{s})\pi_t(\boldsymbol{a}|\boldsymbol{s}) - d_{\mathcal{A},t}(\boldsymbol{s})\pi_t(\boldsymbol{a}|\boldsymbol{s})\right) r_t(\boldsymbol{s},\boldsymbol{a}) \\
&\le \sum_{t=1}^{T}\sum_{\boldsymbol{s}\in S}|d_{\pi_t}(\boldsymbol{s}) - d_{\mathcal{A},t}(\boldsymbol{s})|,
\end{aligned}
$$

where $d_{\mathcal{A},t}(\boldsymbol{s})$ is the state distribution at time step t by following the policy sequence $\pi_1,\ldots,\pi_t$ generated by the OMDP-PI algorithm. The last line can be obtain by $r_t(\boldsymbol{s},\boldsymbol{a}) \in [0,1], \forall t = 1,\ldots,T$.

For all $k = 2,\ldots,t$, we have following results

$$
\begin{aligned}
&\|d_{\mathcal{A},k} - d_{\pi_t}\|_1 \\
&= \|d_{\mathcal{A},k-1}P^{\pi_k} - d_{\pi_{t-1}}P^{\pi_t}\|_1 \\
&\le \|d_{\mathcal{A},k-1}P^{\pi_k} - d_{\mathcal{A},k-1}P^{\pi_t}\|_1 + \|d_{\mathcal{A},k-1}P^{\pi_t} - d_{\pi_{t-1}}P^{\pi_t}\|_1 \\
&\le (\ln{(t-1)} - \ln{(k-1)}) + e^{-1/\tau}\|d_{\mathcal{A},k-1} - d_{\pi_{t-1}}\|_1,
\end{aligned}
$$

Recurring the above result, we have

$$\begin{aligned}\|d_{\mathcal{A},t} - d_{\pi_t}\|_1 &\leq \sum_{k=2}^{t} (\ln(t-1) - \ln(k-1)) e^{-(t-k)/\tau} + e^{-t/\tau}\|d_1 - d_{\pi_t}\|_1 \\ &\leq (1+\tau)\left(\frac{\tau^2}{t-1} + \tau e^{-(t-\tau-2)/\tau}\right) + 2e^{-t/\tau},\end{aligned}$$

where the last inequality follows by

$$\begin{aligned}&\sum_{k=2}^{t} (\ln(t-1) - \ln(k-1)) e^{-(t-k)/\tau} \\ &= \int_2^t (\ln(t-1) - \ln(k-1)) e^{-(t-k)/\tau} \mathrm{d}k + \ln(t-1) e^{-(t-2)/\tau} \\ &= \tau \int_2^t (\ln(t-1) - \ln(k-1)) \frac{\mathrm{d}e^{-(t-k)/\tau}}{\mathrm{d}k} \mathrm{d}k + \ln(t-1) e^{-(t-2)/\tau} \\ &\leq \tau \int_2^t \frac{e^{-(t-k)/\tau}}{k-1} \mathrm{d}k = \tau^2 \int_2^t \frac{1}{k-1} \frac{\mathrm{d}e^{-(t-k)/\tau}}{\mathrm{d}k} \mathrm{d}k \\ &\leq \frac{\tau^2}{t-1} + \tau^2 \int_2^t \frac{e^{-(t-k)/\tau}}{(k-1)^2} \mathrm{d}k \\ &= \frac{\tau^2}{t-1} + \tau^2 \int_{\tau+2}^t \frac{e^{-(t-k)/\tau}}{(k-1)^2} \mathrm{d}k + \int_2^{\tau+2} \frac{e^{-(t-k)/\tau}}{(k-1)^2} \mathrm{d}k \\ &\leq \frac{\tau^2}{t-1} + \frac{\tau^2}{\tau+1} \int_2^t \frac{e^{-(t-k)/\tau}}{k-1} \mathrm{d}k + \int_2^{\tau+2} \frac{e^{-(t-k)/\tau}}{(k-1)^2} \mathrm{d}k.\end{aligned}$$

Hence, we have

$$\int_2^t \frac{e^{-(t-k)/\tau}}{k-1} \mathrm{d}k \leq \left(1 + \frac{1}{\tau}\right)\left(\frac{\tau^2}{t-1} + \tau e^{-(t-\tau-2)/\tau}\right).$$

The claimed result in Lemma 4.4 can be obtained as

$$\sum_{t=1}^{T} \|d_{\mathcal{A},t} - d_{\pi_t}\|_1 \leq 2\tau^3 \ln T + 2\tau^3 + 2\tau^3 e^{(\tau+2)} + 2\tau.$$

4.6.4 Proof of Proposition 4.6

Proposition 4.7. *For arbitrary reward function $r(\boldsymbol{s}, \boldsymbol{a})$, the corresponding value functions induced by two arbitrary policy π_1 and π_2 satisfy*

$$\|\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}\|_\infty \leq C_\pi \|\pi_1 - \pi_2\|_1.$$

where C_π is a positive constant.

Let us define an auxiliary sequence of functions $\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}), t = 1, \ldots, T$ which is defined as

$$\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) = \mathbb{E}_{\pi_t^*}\left[\sum_{i=1}^{\infty}(\hat{r}_t(\boldsymbol{s}, \boldsymbol{a}) - \rho_{\hat{r}_t}(\pi_t^*))\right].$$

In above definition, π_t^* is the optimal policy which satisfies

$$\pi_t^* = \mathrm{argmax}_{\pi \in \Pi}\, \rho_{\hat{r}_t}(\pi),$$

and for all $\boldsymbol{s} \in S$, there is

$$\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) \geq \mathcal{V}_{\hat{r}_t}^{\pi}(\boldsymbol{s}), \forall \pi \in \Pi.$$

It is simple to verify that the value function is linear with respect to the reward function, i.e., $\mathcal{V}_{\hat{r}_t}^{\pi}(\boldsymbol{s}) = \frac{1}{t}\sum_{k=1}^{t}\mathcal{V}_{r_t}^{\pi}(\boldsymbol{s})$. Hence, we can rewrite the sequence as

$$\begin{aligned}
\mathcal{V}_{\hat{r}_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) &= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\left(\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_{t+1}^*}(\boldsymbol{s}) - \frac{t+1}{t}\sum_{k=1}^{t}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s})\right) \\
&= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\mathcal{V}_{r_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) - \frac{1}{t(t+1)}\sum_{k=1}^{t}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s}) \\
&+ \frac{1}{t+1}\left(\sum_{k=1}^{t}\mathcal{V}_{r_k}^{\pi_{t+1}^*}(\boldsymbol{s}) - \sum_{k=1}^{t}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s})\right) \\
&= (1 - \frac{1}{t+1})\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\mathcal{V}_{r_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) + \frac{t}{t+1}\left(\mathcal{V}_{\hat{r}_t}^{\pi_{t+1}^*}(\boldsymbol{s}) - \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s})\right) \\
&\leq (1 - \frac{1}{t+1})\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\mathcal{V}_{r_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}),
\end{aligned}$$

where the last inequality can be obtained by the fact that π_t^* is the optimal policy satisfies

$$\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) \geq \mathcal{V}_{\hat{r}_t}^{\pi_{t+1}^*}(\boldsymbol{s}), \forall \boldsymbol{s} \in S.$$

On the other hand, we can derive the lower bound as

$$\begin{aligned}
\mathcal{V}_{\hat{r}_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) &= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\left(\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_{t+1}^*}(\boldsymbol{s}) - \frac{t+1}{t}\sum_{k=1}^{t}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s})\right)\\
&= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\left(\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_{t+1}^*}(\boldsymbol{s}) - \frac{t+1}{t}\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s})\right) + \frac{1}{t}\mathcal{V}_{r_{t+1}}^{\pi_t^*}(\boldsymbol{s})\\
&= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\left(\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_{t+1}^*}(\boldsymbol{s}) - \sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s})\right)\\
&\quad - \frac{1}{t(t+1)}\sum_{k=1}^{t+1}\mathcal{V}_{r_k}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t}\mathcal{V}_{r_{t+1}}^{\pi_t^*}(\boldsymbol{s})\\
&= \mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + (\mathcal{V}_{\hat{r}_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) - \mathcal{V}_{\hat{r}_{t+1}}^{\pi_t^*}(\boldsymbol{s})) - \frac{1}{t+1}\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\mathcal{V}_{r_{t+1}}^{\pi_t^*}(\boldsymbol{s})\\
&\geq (1 - \frac{1}{t+1})\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) + \frac{1}{t+1}\mathcal{V}_{r_{t+1}}^{\pi_t^*}(\boldsymbol{s}),
\end{aligned}$$

where the last inequality comes from the fact π_{t+1}^* is the optimal policy satisfies

$$\mathcal{V}_{\hat{r}_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) \geq \mathcal{V}_{\hat{r}_{t+1}}^{\pi_t^*}(\boldsymbol{s}), \forall \boldsymbol{s} \in S.$$

Then, we can obtain the following result

$$|\mathcal{V}_{\hat{r}_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) - V_{t+1}(\boldsymbol{s})| \leq (1 - \frac{1}{t+1})|\mathcal{V}_{\hat{r}_t}^{\pi_t^*}(\boldsymbol{s}) - V_t(\boldsymbol{s})| + \frac{1}{t+1}\Delta_{t+1}. \tag{4.7}$$

In above inequality, $\Delta_{t+1} = \max\{|\mathcal{V}_{r_{t+1}}^{\pi_{t+1}^*}(\boldsymbol{s}) - V_{r_{t+1}}^{\pi_{t+1}}(\boldsymbol{s})|, |\mathcal{V}_{r_{t+1}}^{\pi_t^*}(\boldsymbol{s}) - V_{r_{t+1}}^{\pi_{t+1}}(\boldsymbol{s})|\}$, which satisfies

$$\begin{aligned}
\Delta_{t+1} \leq& C_\pi \max\{\|\pi_{t+1}^* - \pi_{t+1}\|_1, \|\pi_t^* - \pi_{t+1}\|_1\}\\
\leq& C_\pi(\|\pi_t^* - \pi_{t+1}\|_1 + \|\pi_t^* - \pi_{t+1}^*\|_1)\\
\leq& C_v\|\mathcal{V}_{\hat{r}_t}^{\pi_t^*} - V_t\|_\infty + \frac{(4\tau+2)C_v}{t+1}.
\end{aligned}$$

The first term of the last inequality can be obtain by setting $C_v = \xi C_\pi$. The second part follows by the upper bound and the lower bound of $\mathcal{V}_{\hat{r}_t}^{\pi_{t+1}^*}$. Next we show the bound of $\|\mathcal{V}_{\hat{r}_t}^{\pi_t^*} - V_t\|_\infty$ by recurring Equ.(4.7)

$$\|\mathcal{V}_{\hat{r}_t}^{\pi_t^*} - V_t\|_\infty \leq \frac{(4\tau+2)C_v}{t^2} + \sum_{k=1}^{t-1}\frac{(4\tau+2)C_v}{k^2}\prod_{m=k+1}^{t}\left(1 - \frac{1-C_v}{m}\right).$$

Let us take the logarithm of $\prod_{m=k+1}^{t}\left(1-\frac{1-C_v}{m}\right)$, there is

$$\begin{aligned}
&\ln\prod_{m=k+1}^{t}\left(1-\frac{1-C_v}{m}\right)\\
&=\sum_{m=k+1}^{t}\left(\ln\left(m-1+C_v\right)-\ln m\right)\\
&\leq\sum_{m=k+1}^{t}\frac{-1+C_v}{m}\\
&\leq-(1-C_v)\int_{k+1}^{t+1}\frac{1}{m}\mathrm{d}m=-(1-C_v)\ln\frac{t+1}{k+1},
\end{aligned}$$

where the first inequality holds since the logarithm function is concave. Thus we derive the bound as

$$\begin{aligned}
\|\mathcal{V}_{\hat{r}_t}^{\pi_t^*}-V_t\|_\infty\leq&(4\tau+2)C_v\sum_{k=1}^{t}\frac{1}{k^2}\frac{(k+1)^{1-C_v}}{(t+1)^{1-C_v}}\\
\leq&\frac{(4\tau+2)C_v}{(t+1)^{1-C_v}}\sum_{k=1}^{t}\frac{1}{k^2}\left(k^{1-C_v}+(1-C_v)k^{-C_v}\right)\\
\leq&\frac{(4\tau+2)C_v}{(t+1)^{1-C_v}}\left[2-C_v+\int_1^t k^{-C_v-1}\mathrm{d}k+(1-C_v)\int_1^t k^{-C_v-2}\mathrm{d}k\right]\\
=&\frac{(4\tau+2)C_v}{(t+1)^{1-C_v}}\left[2-C_v+\frac{1}{C_v}-\frac{t^{-C_v}}{C_v}+\frac{1-C_v}{1+C_v}-\frac{1-C_v}{1+C_v}t^{-C_v-1}\right]\\
\leq&CC_v(t+1)^{C_v-1},
\end{aligned}$$

where $C=6\tau(2-C_v+\frac{1}{C_v}+\frac{1-C_v}{1+C_v})$. In above results, the second inequality follows by Taylor's theorem.

4.6.5 Proof of Proposition 4.7

Let us define the operator

$$\mathcal{T}^\pi\mathcal{V}_r^\pi(\boldsymbol{s})=\mathbb{E}_\pi\left[r(\boldsymbol{s},\boldsymbol{a})-\rho_r(\pi)+\sum_{\boldsymbol{s}'\in S}p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^\pi(\boldsymbol{s}')\right].$$

Then we can obtain

$$\begin{aligned}
&\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{V}_r^{\pi_2}(\boldsymbol{s}) \\
&= \mathcal{T}^{\pi_1}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_2}(\boldsymbol{s}) \\
&= \left(\mathcal{T}^{\pi_1}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_1}(\boldsymbol{s})\right) + \left(\mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_2}(\boldsymbol{s})\right).
\end{aligned}$$

By the definition of the operator, we rewrite the first term as

$$\begin{aligned}
&\mathcal{T}^{\pi_1}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) \\
&= \mathbb{E}_{\pi_1}\left[r(\boldsymbol{s},\boldsymbol{a}) - \rho_r(\pi_1) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^{\pi_1}(\boldsymbol{s}')\right] \\
&\quad - \mathbb{E}_{\pi_2}\left[r(\boldsymbol{s},\boldsymbol{a}) - \rho_r(\pi_2) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^{\pi_1}(\boldsymbol{s}')\right] \\
&= \left(\mathbb{E}_{\pi_1}\left[Q_r^{\pi_1}(\boldsymbol{s},\boldsymbol{a})\right] - \mathbb{E}_{\pi_2}\left[Q_r^{\pi_1}(\boldsymbol{s},\boldsymbol{a})\right]\right) + \left(\rho_r(\pi_2) - \rho_r(\pi_1)\right) \\
&= \left(Q_r^{\pi_1}(\boldsymbol{s},\pi_1) - Q_r^{\pi_1}(\boldsymbol{s},\pi_2)\right) + \mathbb{E}_{\boldsymbol{s}\sim d_{\pi_2}(\boldsymbol{s})}[Q_r^{\pi_1}(s,\pi_2) - Q_r^{\pi_1}(s,\pi_1)].
\end{aligned}$$

The second term can be expressed as

$$\begin{aligned}
&\mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{T}^{\pi_2}\mathcal{V}_r^{\pi_2}(\boldsymbol{s}) \\
&= \mathbb{E}_{\pi_2}\Bigg[r(\boldsymbol{s},\boldsymbol{a}) - \rho_r(\pi_2) + \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^{\pi_1}(\boldsymbol{s}') - r(\boldsymbol{s},\boldsymbol{a}) \\
&\qquad + \rho_r(\pi_2) - \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\mathcal{V}_r^{\pi_2}(\boldsymbol{s})\Bigg] \\
&= \mathbb{E}_{\boldsymbol{s}'\sim p^{\pi_2}(\boldsymbol{s}'|\boldsymbol{s})}[\mathcal{V}_r^{\pi_1}(\boldsymbol{s}') - \mathcal{V}_r^{\pi_2}(\boldsymbol{s}')].
\end{aligned}$$

By summing up the above results, we obtain

$$\begin{aligned}
&\mathcal{V}_r^{\pi_1}(\boldsymbol{s}) - \mathcal{V}_r^{\pi_2}(\boldsymbol{s}) \\
&= \left(Q_r^{\pi_1}(\boldsymbol{s},\pi_1) - Q_r^{\pi_1}(\boldsymbol{s},\pi_2)\right) + \mathbb{E}_{\boldsymbol{s}\sim d_{\pi_2}(\boldsymbol{s})}[Q_r^{\pi_1}(s,\pi_2) - Q_r^{\pi_1}(s,\pi_1)] \\
&\quad + \mathbb{E}_{\boldsymbol{s}'\sim p^{\pi_2}(\boldsymbol{s}'|\boldsymbol{s})}[\mathcal{V}_r^{\pi_1}(\boldsymbol{s}') - \mathcal{V}_r^{\pi_2}(\boldsymbol{s}')].
\end{aligned}$$

In matrix notation, there is

$$\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2} = (Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}) - \boldsymbol{e}_{|S|}d_{\pi_2}^\top(Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}) + P^{\pi_2}(\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}),$$

where $\mathcal{V}_r^\pi$ and $Q_r^{\pi,\pi'}$ are the length $|S|$ vectors whose $\boldsymbol{s}$th element is $\mathcal{V}_r^\pi(\boldsymbol{s})$ and $Q_r^\pi(\boldsymbol{s},\pi')$, respectively. $\boldsymbol{e}_{|S|}$ denotes the length $|S|$ vector with all elements equal to 1. d_π is the $|S|$-dimensional vector whose $\boldsymbol{s}$th element is $d_\pi(\boldsymbol{s})$. P^π is defined as the transition matrix induced by the policy π and the transition $p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$. Thus, we obtain

$$(\boldsymbol{I}_{|S|} - P^{\pi_2})(\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}) = (\boldsymbol{I}_{|S|} - \boldsymbol{e}_{|S|}d_{\pi_2}^\top)(Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}).$$

It is known that the Bellman equation with average reward function has no unique solution. However, the unique value function satisfies $d_\pi^\top \mathcal{V}_r^\pi = 0$. Hence, we add this condition to the above equation as

$$\begin{aligned}
&(\boldsymbol{I}_{|S|} - P^{\pi_2})(\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2})\\
&= (\boldsymbol{I}_{|S|} - \boldsymbol{e}_{|S|}d_{\pi_2}^\top)(Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}) - \boldsymbol{e}_{|S|}d_{\pi_1}^\top \mathcal{V}_r^{\pi_1} + \boldsymbol{e}_{|S|}d_{\pi_2}^\top \mathcal{V}_r^{\pi_2}\\
&= (\boldsymbol{I}_{|S|} - \boldsymbol{e}_{|S|}d_{\pi_2}^\top)(Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}) - \boldsymbol{e}_{|S|}d_{\pi_2}^\top(\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2})\\
&\quad - (\boldsymbol{e}_{|S|}d_{\pi_1}^\top - \boldsymbol{e}_{|S|}d_{\pi_2}^\top)\mathcal{V}_r^{\pi_1}
\end{aligned}$$

Then, by rearranging the above result:

$$(\boldsymbol{I}_{|S|} - P^{\pi_2} + \boldsymbol{e}_{|S|}d_{\pi_2}^\top)(\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}) = (Q_r^{\pi_1,\pi_1} - Q_r^{\pi_1,\pi_2}) - (P_{sa}^{\pi_1} - P_{sa}^{\pi_2})Q_r^{\pi_1},$$

where P_{sa}^π is the $|S| \times |A|$ matrix whose $(\boldsymbol{s},\boldsymbol{a})$th element is $d^\pi(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s})$. Using Proposition 12 in Ma et al. (2014), we obtain

$$\|\mathcal{V}_r^{\pi_1} - \mathcal{V}_r^{\pi_2}\|_\infty \leq \frac{2 - 2e^{-1/\tau}}{1 - e^{-1/\tau}} \|(\boldsymbol{I}_{|S|} - P^{\pi_2} + \boldsymbol{e}_{|S|}d_{\pi_2}^\top)^{-1} Q_r^{\pi_1}\|_\infty \|\pi_1 - \pi_2\|_1,$$

which concludes the proof by setting

$$\max_{\pi\in\Pi} \frac{2 - 2e^{-1/\tau}}{1 - e^{-1/\tau}} \|(\boldsymbol{I}_{|S|} - P^\pi + \boldsymbol{e}_{|S|}d_\pi^\top)^{-1} Q_r^\pi\|_\infty \leq C_\pi.$$

4.7 Summary

As a generalization of MDP, online MDP is a promising model which can handle many online problem with guaranteed performance. In this chapter, we proposed a policy iteration algorithm with a closed form update rule for online MDP problems. We showed that the proposed algorithm achieves sublinear regret for a policy set. A notable fact is that the proposed algorithm is still practical for online MDP problems with large (continuous) state space. We showed that the propose algorithm can be easily combined with function approximation with theoretical guarantee. We illustrated the performance of the proposed algorithm through grid-world experiments.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Online decision making in non-stationary Markovian environments has a promising prospect of application from robotics to finance. By formulating the problem as an online MDP, a growing number of researches proposed online MDP algorithms and analyzed their online performance with theoretical guarantees. Particularly, online MDP problems with large (continuous) state space arise naturally. In this thesis, we proposed two new algorithms for dealing with such problems. As shown in Figure 1.3, the proposed algorithms handle the continuous online MDP problem by parameterizing the policy space and the value function, respectively.

In Chapter 3, we proposed the online policy gradient algorithm for continuous state and action spaces online MDP problems. The proposed algorithm utilizes the parameterized policy model which is natural for handling continuous state and action spaces. Through regret analysis, we showed that the proposed algorithm achieves a sublinear regret, which means our algorithm performs asymptotically equal to the best fixed policy. More precisely, the proposed algorithm achieves $O(\sqrt{T})$ and $O(\log T)$ regrets with full information feedback under concavity and strong concavity assumptions, respectively. Furthermore, the proposed algorithm achieves $O(\sqrt{T})$ with bandit feedback under a concavity assumption. We also demonstrated the performance of our algorithm with two toy experiments, which verifies that our algorithm improves the performance for continuous tasks.

In Chapter 4, we proposed the online MDPs with policy iteration algorithm for large (infinite) state space online MDP problems, which achieves less computation complexity in exchange for large regret. The proposed algorithm is motivated by the idea of combining the function approximation with policy iteration. Through regret analysis, we proved that the proposed method achieve a sublinear regret with full information feedback. We also analyzed the computation complexity is $O(|S|^{2.3728639} + |S|^2|A|)$, which is computational more efficient than related works. Then we presented a linear approximator with a convergence guarantee, which can be used together with the OMDP-PI algorithm. Moreover, an extension of the OMDP-PI algorithm called OMDP-SI algorithm is presented. We showed that the OMDP-SI algorithm could achieve a sublinear regret as well under some specific additional assumptions. Finally, we illustrated the experimental performance to show the usefulness of the OMDP-PI algorithm.

5.2 Future Work

In this section, we show some challenging directions that we will work on in the future.

5.2.1 Non-stationary Transition Dynamics

In an online MDP, the environment consists of two components: the reward function and transition dynamics. In this thesis, we assumed the reward function changes over time and transition dynamics are fixed and known to the decision maker. However, this is not the best we can do with changing environments. A more challenging problem is that the decision maker faces two kinds of uncertainties of the environment: the changing reward function and the changing transition probability (density).

A related work to handle this uncertainty is robust optimization. Plenty of algorithms have been proposed to handle robust Markov decision processes, where the reward function and the transition dynamics are allowed to change in a certain range. These algorithms usually assumes that the uncertainty has a fixed unknown realization and construct a policy that performs well in the worst case (Yu et al., 2009). In other words, the robust optimization gives a policy which performs reasonably well within the set of all the reward functions and transition probabilities as

$$\pi^{+} = \mathrm{argmax}_{\pi} \min_{(r,p)\in\mathcal{U}} \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)|p\right],$$

where $\mathcal{U}$ is the set of all the reward functions and transition probabilities pairs.

The robust optimization cannot face the arbitrary changing environment as we considered in online MDPs. Yu and Mannor (2009) proposed an algorithm that combines the online MDP and the robust optimization by assuming the transition dynamics are not changing abruptly. Nevertheless, this assumption does not always hold in reality. It is challenging and important to find a good way to solve the changing transition dynamics problem without additional assumptions.

5.2.2 Contextual Online MDPs

Recently, Abbasi-Yadkori and Neu (2014) considered online Markov decision process problems when the side information are available. This problem is motivated by real applications where the environments are usually not arbitrary changing over time. The recommendation system we introduced in Section 1.3 is a typical example of online MDP problem with side information. Consider the recommendation system decides the recommendations which should be provided to the customers depending on the users' profiles. Every user could be specified by the user's private information, which can be treated as the side information. The contextual online MDP problem is similar to the contextual bandit problem. The environment change is neither stochastic nor adversary, where the reward function and the transition dynamic are allowed to depend on the side information.

Abbasi-Yadkori and Neu (2014) defined that the reward function and the transition probability are both the generalized linear model with respect to the side information vector. The idea is to maintain confidence sets for the parameters of the generalized linear models, then choose the estimation and the policy which maximizes the return.

However, this approach is applicable only to the generalized linear model. Therefore, it is a promising direction to investigate a more general way to solve contextual online MDP problems.

5.2.3 Beyond the Regret

Till now, we used the notion of the regret for evaluating the performance of online MDP algorithms. However, the regret is not the only way to evaluate the online performance. Beyond the regret, other performance measure (e.g., Φ-regret, internal regret) can be stronger than the regret. Consider the baseline of comparison, a more challenging baseline is possible to be proposed for the definition of the regret. Yu et al. (2009) considered the regret with respect to dynamic policies over T time steps as

$$\tilde{L}_{\mathcal{A}}(T) = \sup_{\{\pi_1,\ldots,\pi_T\}} \mathbb{E}_{\{\pi_1,\ldots,\pi_T\}} \left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] - R_{\mathcal{A}}(T),$$

where the switching times of the policy sequence $_{\{\pi_1,\ldots,\pi_T\}}$ are bounded by some positive constant. With addition assumptions, it has been shown that the above definition of the regret can be sublinear in online MDP problems.

Therefore, we would like to investigate a stronger regret bound for our proposed algorithm in future works.

Bibliography

Yasin Abbasi-Yadkori and Gergely Neu. Online learning in MDPs with side information. *arXiv preprint arXiv:1406.6812*, 2014.

Yasin Abbasi-Yadkori, Peter Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems 26*, pages 2508–2516. 2013.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3): 167–175, 2003.

Richard Bellman. A Markovian decision process. Technical report, DTIC Document, 1957.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996. ISBN 1886529108.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5, 2012.

Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*, volume 39. CRC press, 2010.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Balázs Csanád Csáji and László Monostori. Value function based reinforcement learning in changing Markovian environments. *Journal of Machine Learning Research*, 9:1679–1709, June 2008. ISSN 1532-4435.

Travis Dick, András György, and Csaba Szepesvári. Online learning in Markov decision processes with changing cost sequences. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 512–520, 2014.

Eyal Even-Dar, Sham M. Kakade, and Yishay Mansour. Experts in a Markov decision process. In *Advances in Neural Information Processing System*, pages 401–408, 2003.

Eyal Even-Dar, Sham M. Kakade, and Yishay Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0-89871-585-7. URL http://dl.acm.org/citation.cfm?id=1070432.1070486.

James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3(97-139):2, 1957.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007. ISSN 0885-6125.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, October 2005. ISSN 0022-0000.

Arpad Kelemen, Yulan Liang, and Stan Franklin. A comparative study of different machine learning approaches for decision making. In *Recent Advances in Simulation, Computational Methods and International Journal of Computer Science Issues*, 2002.

Robert David Kleinberg. *Online decision problems with large strategy sets*. PhD thesis, Citeseer, 2005.

François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2501-1.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In *Advances in Neural Information Processing Systems 24*, pages 19–27, 2011.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Yao Ma, Tingting Zhao, Kohei Hatano, and Masashi Sugiyama. An online policy gradient algorithm for Markov decision processes with continuous states and actions. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, pages 354–369, 2014.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Gergely Neu, András György, and Csaba Szepesvári. The online loop-free stochastic shortest-path problem. In *The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 231–243, 2010a.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing System,NIPS*, pages 1804–1812, 2010b.

Gergely Neu, András György, and Csaba Szepesvári. The adversarial stochastic shortest path problem with unknown transition probabilities. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, volume 22, pages 805–813, 2012.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online Markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691, March 2014.

Andrew Y. Ng, Ronald Parr, and Daphne Koller. Policy search via density estimation. In *In Advances in Neural Information Processing Systems 12*, pages 1022–1028. MIT Press, 1999.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press, 2007.

Ronald Edward Parr. *Hierarchical control and learning for Markov decision processes*. PhD thesis, Citeseer, 1998.

Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225, Oct 2006.

James Renegar. Some perturbation theory for linear programming. *Mathematical Programming*, 65(1):73–91, 1994.

Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, pages 9–44, 1988.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.

Istvan Szita, Bálint Takács, and András Lörincz. MDPs: Learning in varying environments. *Journal of Machine Learning Research*, 3:145–174, 2002.

John N. Tsitsiklis and Benjamin Van Roy. Average cost temporal-difference learning. *Automatica*, 35:1799–1808, 1999.

Volodimir G Vovk. Aggregating strategies. In *Proceedings of the Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992. ISSN 0885-6125. doi: 10.1007/BF00992696.

Huizhen Yu. *Approximate solution methods for partially observable Markov and semi-Markov decision processes*. PhD thesis, Massachusetts Institute of Technology, 2006.

Jia Yuan Yu and Shie Mannor. Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *International Conference on Game Theory for Networks, 2009. GameNets' 09.*, pages 314–322. IEEE, 2009.

Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Alexander Zimin and Gergely Neu. Online learning in episodic Markovian decision processes by relative entropy policy search. In *Advances in Neural Information Processing Systems 26*, pages 1583–1591, 2013.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning ICML*, pages 928–936, 2003.