Learning to Collaborate in Markov Decision Processes

Goran Radanovic [1] **Rati Devidze** [2] **David C. Parkes** [1] **Adish Singla** [2]

Abstract

We consider a two-agent MDP framework where agents repeatedly solve a task in a collaborative setting. We study the problem of designing a learning algorithm for the first agent ($\mathcal{A}_1$) that facilitates successful collaboration even in cases when the second agent ($\mathcal{A}_2$) is adapting its policy in an unknown way. The key challenge in our setting is that the first agent faces non-stationarity in rewards and transitions because of the adaptive behavior of the second agent.

We design novel online learning algorithms for agent $\mathcal{A}_1$ whose regret decays as $\mathcal{O}\Big(T^{\max\{1-\frac{3}{7}\cdot\alpha,\frac{1}{4}\}}\Big)$, for T learning episodes, provided that the magnitude in the change in agent $\mathcal{A}_2$'s policy between any two consecutive episodes is upper bounded by $\mathcal{O}(T^{-\alpha})$. Here, the parameter α is assumed to be strictly greater than 0, and we show that this assumption is necessary provided that the *learning parity with noise* problem is computationally hard. We show that sublinear regret of agent $\mathcal{A}_1$ further implies near-optimality of the agents' joint return for MDPs that manifest the properties of a *smooth* game.

1. Introduction

Recent advancements in AI have the potential to change our daily lives by boosting productivity (e.g., via virtual personal assistants), augmenting human capabilities (e.g., via smart mobility systems), and increasing automation (e.g., via auto-pilots and assistive robots). These are settings of intelligence augmentation, where societal benefit will come not from complete automation but rather from the interaction between people and machines, in a process of a productive human-machine collaboration.

[1]Harvard University, School of Applied Science and Engineering. [2]The Max Planck Institute for Software Systems. Correspondence to: Goran Radanovic <gradanovic@g.harvard.edu>, Adish Singla <adishs@mpi-sws.org>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

We expect that useful collaboration will come about through AI agents that can adapt to the behavior of users. As an example, consider self-driving cars where auto-pilots can be overridden by human drivers. In an initial period, a human driver would likely change their behavior until accustomed with new features of an auto-pilot mode. Without accounting for this changing behavior of users, the performance of the AI agent could considerably deteriorate, leading to, for example, hazardous situations in an auto-pilot mode. Hence, it is important that the AI agent updates its decision-making policy accordingly.

We formalize this problem through a two-agent, reinforcement learning (RL) framework. The agents, hereafter referred to as agent $\mathcal{A}_1$ and agent $\mathcal{A}_2$, jointly solve a task in a collaborative setting (i.e., share a common reward function and a transition kernel that is based on their joint actions). Our goal is to develop a learning algorithm for agent $\mathcal{A}_1$ that facilitates a successful collaboration even in cases when agent $\mathcal{A}_2$ is adapting its own policy. In the above examples, agent $\mathcal{A}_1$ could represent the AI agent whereas agent $\mathcal{A}_2$ could be a person with time-evolving behavior. We primarily focus on an *episodic* Markov decision process (MDP) setting, in which the agents repeatedly interact:

(i) agent $\mathcal{A}_1$ decides on its policy based on historic information (agent $\mathcal{A}_2$'s past policies) and the underlying MDP model;

(ii) agent $\mathcal{A}_1$ commits to its policy for a given episode without knowing the policy of agent $\mathcal{A}_2$;

(iii) agent $\mathcal{A}_1$ updates its policy at the end of the episode based on agent $\mathcal{A}_2$'s observed behavior.

When agent $\mathcal{A}_2$'s policy is fixed and known, one can find an optimal policy for agent $\mathcal{A}_1$ using standard MDP planning techniques. In our setting, however, we do not assume agent $\mathcal{A}_2$'s behavior is stationary, and we do not assume any particular model for how agent $\mathcal{A}_2$ changes its policy. This differs from similar two-agent (human-AI) collaborative settings (Dimitrakakis et al., 2017; Nikolaidis et al., 2017) that prescribe a particular behavioral model to agent $\mathcal{A}_2$ (human agent).

1.1. Overview of Our Approach

The presence of agent $\mathcal{A}_2$ in our framework implies that the reward function and transition kernel are non-stationary

from the perspective of agent $\mathcal{A}_1$. Variants of the setting have also been studied in the learning literature (Even-Dar et al., 2005; 2009; Yu & Mannor, 2009b;a; Yu et al., 2009; Abbasi et al., 2013; Wei et al., 2017). However, these approaches do not directly apply because: (i) they focus on a particular aspect of non-stationarity (e.g., changing rewards with fixed transitions) (Even-Dar et al., 2005; 2009), (ii) require that the changes in the transition model are bounded (Yu & Mannor, 2009a;b), (iii) make restrictions on the policy space (Abbasi et al., 2013), and (iv) consider a competitive or adversarial setting instead of cooperative setting with shared reward (Wei et al., 2017). Instead, we will assume that agent $\mathcal{A}_2$ does not abruptly change its policy across episodes, and prove that the problem becomes computationally intractable otherwise. Our approach is inspired by the problem of experts learning in MDPs (Even-Dar et al., 2005), in which each state is associated with an experts algorithm that derives the policy for that state using Q-values. However, to compensate for the non-stationarity of transitions and facilitate a faster learning process, we introduce novel forms of recency bias inspired by the ideas of Rakhlin & Sridharan (2013); Syrgkanis et al. (2015).

Contributions. We design novel algorithms for agent $\mathcal{A}_1$ that lead to sub-linear regret of $\mathcal{O}(T^{\max\{1-\frac{3}{7}\cdot\alpha,\frac{1}{4}\}})$, where T is the number of episodes. We assume the magnitude of agent $\mathcal{A}_2$'s policy change w.r.t. T is $\mathcal{O}(T^{-\alpha})$, for parameter α that we require to be strictly positive. We show via a reduction from the *learning parities with noise* problem (Abbasi et al., 2013; Kanade & Steinke, 2014) that this upper bound on the rate of change in agent $\mathcal{A}_2$'s policy is necessary, in that it is computationally hard to achieve sub-linear regret for the special case of $\alpha = 0$. Furthermore, we connect the agents' joint return to the regret of agent $\mathcal{A}_1$ by adapting the concept of *smoothness* from the game-theory literature (Roughgarden, 2009; Syrgkanis et al., 2015), and we show that the bound on the regret of agent $\mathcal{A}_1$ implies near optimality of the agents' joint return for MDPs that manifest a *smooth* game (Roughgarden, 2009; Syrgkanis et al., 2015). To the best of our knowledge, we are the first to provide such guarantees in a collaborative two-agent MDP learning setup. The proofs can be found in the extended version of the paper (Radanovic et al., 2019).

2. The Setting

We model a two-agent learning problem through an MDP environment.[1] The agents are agent $\mathcal{A}_1$ and agent $\mathcal{A}_2$. We consider an episodic setting with T episodes (also called time steps) and each episode lasting M rounds. Generic episodes are denoted by t and τ, while a generic round is denoted by m. The MDP is defined by:

- a finite set of states S, with s denoting a generic state. We enumerate the states by 1, ..., $|S|$, and assume this ordering in our vector notation.
- a finite set of actions $A = A_1 \times A_2$, with $a^1 \in A_1$ denoting a generic action of agent $\mathcal{A}_1$ and $a^2 \in A_2$ denoting a generic action of agent $\mathcal{A}_2$. We enumerate the actions of agent $\mathcal{A}_1$ by 1, ..., $|A_1|$ and agent $\mathcal{A}_2$ by 1, ..., $|A_2|$, and assume this ordering in our vector notation.
- a transition kernel $\boldsymbol{P}(s, a^1, a^2, s_{new})$, which is a tensor with indices defined by the current state, the agents' actions, and the next state.
- a reward function $r : S \times A \to [0, 1]$ that defines the joint reward for both agents.

We assume that agent $\mathcal{A}_1$ knows the MDP model. The agents *commit* to playing *stationary* policies ${\boldsymbol{\pi}^1}_t$ and ${\boldsymbol{\pi}^2}_t$ in each episode t, but do so without knowing the commitment of the other agent. At the end of the episode t, the agents observe each other's policies $({\boldsymbol{\pi}^1}_t,\ {\boldsymbol{\pi}^2}_t)$ and can use this information to update their future policies.[2] Since the state and action spaces are finite, policies can be represented as matrices ${\boldsymbol{\pi}^1}_t(s, a^1)$ and ${\boldsymbol{\pi}^2}_t(s, a^2)$, so that rows ${\boldsymbol{\pi}^1}_t(s)$ and ${\boldsymbol{\pi}^2}_t(s)$ define distributions on actions in a given state. We also define the reward matrix for agent $\mathcal{A}_1$ as $\boldsymbol{r}_t(s, a^1) = \mathbb{E}_{a^2 \sim {\boldsymbol{\pi}^2}_t(s)}\left[r(s, a^1, a^2)\right]$, whose elements are the expected rewards of agent $\mathcal{A}_1$ for different actions and states. By bounded rewards, we have $0 \leq \boldsymbol{r}_t(s, a^1) \leq 1$.

2.1. Objective

After each episode t, the agents can adapt their policies. However, agent $\mathcal{A}_2$ is not in our control, and not assumed to be optimal. Therefore, we take the perspective of agent $\mathcal{A}_1$, and seek to optimize its policy in order to obtain good joint returns. The joint return in episode t is:

$$V_t = \frac{1}{M} \cdot \mathbb{E}\left[\sum_{m=1}^{M} r(s_m, {a^1}_m, {a^2}_m) | \boldsymbol{d}_1, {\boldsymbol{\pi}^1}_t, {\boldsymbol{\pi}^2}_t\right]$$
$$= \frac{1}{M} \cdot \sum_{m=1}^{M} \boldsymbol{d}_{t,m} \cdot \left\langle {\boldsymbol{\pi}^1}_t, \boldsymbol{r}_t \right\rangle,$$

where s_m is the state at round m. For $m = 1$, s_m is sampled from the initial state distribution $\boldsymbol{d}_1$. For later periods, s_m is obtained by following joint actions $({a^1}_1, {a^2}_1)$, $({a^1}_2, {a^2}_2)$, ..., $({a^1}_{m-1}, {a^2}_{m-1})$ from state s_1. Actions are obtained from policies ${\boldsymbol{\pi}^1}_t$ and ${\boldsymbol{\pi}^2}_t$. The second part of the equation uses a vector notation to define the joint return, where $\boldsymbol{d}_{t,m}$ is a row vector representing the state dis-

[1] An MDP with multiple agents is often called *multiagent* MDP (Boutilier, 1996).

[2] We focus on the full information setting as it allows us to do a cleaner analysis while revealing some of the challenges of the problem at hand. The setting, for example, formalizes a scenario where the AI describes its policy to the human, and the episodes are long enough that the AI can effectively observe the human's policy.

tribution at episode t and round m, while $\langle {\boldsymbol{\pi}^1}_t, \boldsymbol{r}_t \rangle$ is a row-wise dot product whose result is a column vector with $|S|$ elements. Since this is an episodic framework, we will assume the same starting state distribution, $\boldsymbol{d}_1$, for all episodes t. However $\boldsymbol{d}_{t,m}$ can differ across episodes since policies ${\boldsymbol{\pi}^1}_t$ and ${\boldsymbol{\pi}^2}_t$ evolve.

We define the average return over all episodes as $\bar{V} = \frac{1}{T} \cdot \sum_{t=1}^{T} V_t$. The objective is to output a sequence of agent $\mathcal{A}_1$'s policies ${\boldsymbol{\pi}^1}_1$, ..., ${\boldsymbol{\pi}^1}_T$ that maximize:

$$\sup_{{\boldsymbol{\pi}^1}_1, \ldots, {\boldsymbol{\pi}^1}_T} \bar{V} = \sup_{{\boldsymbol{\pi}^1}_1, \ldots, {\boldsymbol{\pi}^1}_T} \frac{1}{T} \cdot \sum_{t=1}^{T} V_t.$$

The maximum possible value of $\bar{V}$ over all combinations of agent $\mathcal{A}_1$'s and agent $\mathcal{A}_2$'s policies is denoted as OPT. Notice that this value is achievable using MDP planning techniques, provided that we control both agents.

2.2. Policy Change Magnitude and Influences

We do not control agent $\mathcal{A}_2$, and we do note assume that agent $\mathcal{A}_2$ follows a particular behavioral model. Rather, we quantify the allowed behavior via the *policy change magnitude*, which for agent $\mathcal{A}_2$ is defined as:

$$\begin{aligned} \rho_2 &= \max_{t>1, s} \sum_{a^2 \in A_2} |{\boldsymbol{\pi}^2}_t(s, a^2) - {\boldsymbol{\pi}^2}_{t-1}(s, a^2)| \\ &= \max_{t>1} \left\| {\boldsymbol{\pi}^2}_t - {\boldsymbol{\pi}^2}_{t-1} \right\|_\infty , \end{aligned}$$

where $\|\cdot\|_\infty$ is operator (induced) norm. In the case of agent $\mathcal{A}_2$, we will be focusing on policy change magnitudes ρ_2 that are of the order $\mathcal{O}(T^{-\alpha})$, where α is strictly grater than 0. For instance, the assumption holds if agent $\mathcal{A}_2$ is a learning agent that adopts the experts in MDP approach of Even-Dar et al. (2005; 2009).

We also define the *influence* of an agent on the transition dynamics. This measures how much an agent can influence the transition dynamics through its policy. For agent $\mathcal{A}_2$, the influence is defined as:

$$I_2 = \sup_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2 \neq \boldsymbol{\pi}^{2\prime}} \frac{\left\| \boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2} - \boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^{2\prime}} \right\|_\infty}{\left\| \boldsymbol{\pi}^2 - \boldsymbol{\pi}^{2\prime} \right\|_\infty},$$

where kernel (matrix) $\boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2}(s, s_{new})$ denotes the probability of transitioning from s to s_{new} when the agents' policies are $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$ respectively.[3] Influence is a measure of how much an agent affects the transition probabilities by changing its policy. We are primarily interested in using this notion to show how our approach compares to the existing results from the online learning literature.

[3] Our notion of influence is similar to, although not the same as, that of Dimitrakakis et al. (2017).

For $I_2 = 0$, our setting relates to the single agent settings of Even-Dar et al. (2005; 2009); Dick et al. (2014) where rewards are non-stationary but transition probabilities are fixed. In general, the influence I_2 takes values in $[0, 1]$ (see Appendix B (Corollary 1) of Radanovic et al. (2019)). We can analogously define policy change magnitude ρ_1, and influence I_1 of agent $\mathcal{A}_1$.

2.3. Mixing Time and Q-values

We follow standard assumptions from the literature on online learning in MDPs (e.g., see Even-Dar et al. (2005)), and only consider transition kernels that have well-defined stationary distributions. For the associated transition kernel, we define a stationary state distribution $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2}$ as the one for which:

1. any initial state distribution converges to under policies $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$;
2. and $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2} \cdot \boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2} = \boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2}$.

Note that $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2}$ is represented as a row vector with $|S|$ elements. Furthermore, as discussed in Even-Dar et al. (2005), this implies that there exists a mixing time ω, such that for all state distributions $\boldsymbol{d}$ and $\boldsymbol{d}'$, we have

$$\left\| \boldsymbol{d} \cdot \boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2} - \boldsymbol{d}' \cdot \boldsymbol{P}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2} \right\|_1 \leq e^{-\frac{1}{\omega}} \cdot \left\| \boldsymbol{d} - \boldsymbol{d}' \right\|_1 .$$

Due to this well-defined mixing time, we can define the *average reward* of agent $\mathcal{A}_1$ when following policy $\boldsymbol{\pi}^1$ in episode t as:

$$\eta_t(\boldsymbol{\pi}^1) := \eta_{{\boldsymbol{\pi}^2}_t}(\boldsymbol{\pi}^1) = \boldsymbol{d}_{\boldsymbol{\pi}^1, {\boldsymbol{\pi}^2}_t} \cdot \langle \boldsymbol{\pi}^1, \boldsymbol{r}_t \rangle ,$$

where $\langle ., . \rangle$ is row-wise dot product whose result is a column vector with $|S|$ elements. The Q-value matrix for agent $\mathcal{A}_1$ w.r.t. policy ${\boldsymbol{\pi}^1}_t$ is defined as:

$$\mathbf{Q}_t(s, a^1) = \mathbb{E}\left[\sum_{m=1}^{\infty} \left(\boldsymbol{r}_t(s_m, {a^1}_m) - \eta_t({\boldsymbol{\pi}^1}_t) \right) | s, a^1, {\boldsymbol{\pi}^1}_t \right],$$

where s_m and ${a^1}_m$ are states and actions in round m, starting from state s with action a^1 and then using policy ${\boldsymbol{\pi}^1}_t$. Moreover, the policy-wise Q-value (column) vector for $\boldsymbol{\pi}^1$ w.r.t. policy ${\boldsymbol{\pi}^1}_t$ is defined by:

$$\mathbf{Q}_t^{\boldsymbol{\pi}^1}(s) = \mathbb{E}_{a^1 \sim \boldsymbol{\pi}^1(s)} \left[\mathbf{Q}_t(s, a^1) \right],$$

and in matrix notation $\mathbf{Q}_t^{\boldsymbol{\pi}^1} = \langle \boldsymbol{\pi}^1, \mathbf{Q}_t \rangle$. The Q-values satisfy the following Bellman equation:

$$\mathbf{Q}_t(s, a^1) = \boldsymbol{r}_t(s, a^1) - \eta_t({\boldsymbol{\pi}^1}_t) + \boldsymbol{P}_{{\boldsymbol{\pi}^2}_t}(s, a^1) \cdot \mathbf{Q}_t^{{\boldsymbol{\pi}^1}_t},$$

where $\boldsymbol{P}_{{\boldsymbol{\pi}^2}_t}(s, a^1)$ defines the probability distribution over next states given action a^1 of agent $\mathcal{A}_1$ and policy ${\boldsymbol{\pi}^2}_t$ of agent $\mathcal{A}_2$ (here, $\boldsymbol{P}_{{\boldsymbol{\pi}^2}_t}(s, a^1)$ is denoted as a row vector with $|S|$ elements). For other useful properties of this MDP framework we refer the reader to Appendix B of Radanovic et al. (2019).

3. Smoothness and No-regret Dynamics

The goal is to output a sequence of agent $\mathcal{A}_1$'s policies ${\boldsymbol{\pi}^1}_1$, ..., ${\boldsymbol{\pi}^1}_T$ so that the joint return $\bar{V}$ is maximized. There are two key challenges: (i) agent $\mathcal{A}_2$ policies could be sub-optimal (or, even adversarial in the extreme case), and (ii) agent $\mathcal{A}_1$ does not know the current policy of agent $\mathcal{A}_2$ at the beginning of episode t.

Smoothness Criterion. To deal with the first challenge, we consider a structural assumption that enables us to apply a regret analysis when quantifying the quality of a solution w.r.t. the optimum. In particular, we assume that the MDP is *(λ, μ)-smooth*:

Definition 1. *We say that an MDP is* (λ, μ)-smooth *if there exists a pair of policies* $(\boldsymbol{\pi}^{1^*}, \boldsymbol{\pi}^{2^*})$ *such that for every policy pair* $(\boldsymbol{\pi}^1, \boldsymbol{\pi}^2)$*:*

$$\eta_{\boldsymbol{\pi}^2}(\boldsymbol{\pi}^{1^*}) \geq \lambda \cdot \eta_{\boldsymbol{\pi}^{2*}}(\boldsymbol{\pi}^{1^*}) - \mu \cdot \eta_{\boldsymbol{\pi}^2}(\boldsymbol{\pi}^1),$$
$$\eta_{\boldsymbol{\pi}^{2*}}(\boldsymbol{\pi}^{1^*}) \geq \eta_{\boldsymbol{\pi}^2}(\boldsymbol{\pi}^1).$$

This bounds the impact of agent $\mathcal{A}_2$'s policy on the average reward. In particular, there must exist an *optimal* policy pair $(\boldsymbol{\pi}^{1^*}, \boldsymbol{\pi}^{2^*})$ such that the negative impact of agent $\mathcal{A}_2$ for choosing $\boldsymbol{\pi}^2 \neq \boldsymbol{\pi}^{2^*}$ is controllable by an appropriate choice of agent $\mathcal{A}_1$'s policy. This definition is a variant of the *smoothness* notion introduced to study the "price-of-anarchy" of non-cooperative games, including for learning dynamics (Roughgarden, 2009; Syrgkanis et al., 2015). For the relationship between the smoothness parameters and the properties of the MDP, we refer the reader to Appendix C of Radanovic et al. (2019). It is important to note that since we have a finite number of rounds M per episode, OPT is not necessarily the same as $\eta_{\boldsymbol{\pi}^{2*}}(\boldsymbol{\pi}^{1^*})$, and the policies that achieve OPT need not lead to $\eta_{\boldsymbol{\pi}^{2*}}(\boldsymbol{\pi}^{1^*})$.

No-regret Learning. To address the second challenge, we adopt the online learning framework and seek to minimize *regret* $R(T)$:

$$R(T) = \sup_{\boldsymbol{\pi}^1} \sum_{t=1}^{T} \left[\eta_t(\boldsymbol{\pi}^1) - \eta_t({\boldsymbol{\pi}^1}_t)\right]. \tag{1}$$

A policy sequence ${\boldsymbol{\pi}^1}_1$, ..., ${\boldsymbol{\pi}^1}_T$ is *no-regret* if regret $R(T)$ is sublinear in T. An algorithm that outputs such sequences is a *no-regret algorithm* — this intuitively means that the agent's performance is competitive w.r.t. any fixed policy.

Near-optimality of No-regret Dynamics. Because agent $\mathcal{A}_2$ could be adapting to the policies of agent $\mathcal{A}_1$, this is an adaptive learning setting, and the notion of regret can become less useful. This is where the smoothness criterion comes in. We will show that it suffices to minimize the regret $R(T)$ in order to obtain near-optimal performance.

Using an analysis similar to Syrgkanis et al. (2015), we establish the near-optimality of no-regret dynamics defined w.r.t. the optimal return OPT, as stated in the following lemma:

Lemma 1. *For a problem with (λ, μ)-smooth MDP, return $\bar{V}$ is lower bounded by:*

$$\bar{V} \geq \frac{\lambda}{1+\mu} \cdot \text{OPT} - \frac{1}{1+\mu} \cdot \frac{R(T)}{T} - 2 \cdot \frac{1 + \frac{\lambda}{1+\mu}}{M \cdot (1 - e^{-\frac{1}{\omega}})}.$$

Lemma 1 implies that as the number of episodes T and the number of rounds M go to infinity, return $\bar{V}$ converges to a multiple $\frac{\lambda}{1+\mu}$ of the optimum OPT, provided that agent $\mathcal{A}_1$ is a no-regret learner. In the next section, we design such no-regret learning algorithms for agent $\mathcal{A}_1$.

4. Learning Algorithms

We base our approach on the expert learning literature for MDPs, in particular that of Even-Dar et al. (2005; 2009). The basic idea is to associate each state with an experts algorithm, and decide on a policy by examining the Q-values of state-action pairs. Thus, the Q function represents a reward function in the expert terminology.

4.1. Experts with Periodic Restarts: EXPRESTART

In cases when agent $\mathcal{A}_2$ has no influence on transitions, the approach of Even-Dar et al. (2005; 2009) would yield the no-regret guarantee. The main difficulty of the present setting is that agent $\mathcal{A}_2$ can influence the transitions via its policy. The hope is that as long as the magnitude of policy change by agent $\mathcal{A}_2$ across episodes is not too large, agent $\mathcal{A}_1$ can compensate for the non-stationarity by using only recent history when updating its policy.

A simple way of implementing this principle is to use a no-regret learning algorithm, but periodically restarting it, i.e., by splitting the full time horizon into segments of length Γ, and applying the algorithm on each segment separately. In this way, we have well-defined periods $\{1, ..., \Gamma\}$, $\{\Gamma + 1, ..., 2 \cdot \Gamma\}$, ..., $\{T - \Gamma + 1, ..., T\}$. As a choice of an expert algorithm (the algorithm associated with each state), we use *Optimistic Follow the Regularized Leader* (OFTRL) (Rakhlin & Sridharan, 2013; Syrgkanis et al., 2015). Our policy updating rule for segment l, with starting point $\tau = 1 + (l-1) \cdot \Gamma$, can be described as:

$${\boldsymbol{\pi}^1}_t(s) = \arg\max_{\mathbf{w} \in \mathcal{P}_{A_1}} \left(\sum_{k=\tau}^{t-1} \mathbf{Q}_k(s) + \mathbf{Q}_{t-1}(s) \right) \mathbf{w}^\dagger + \frac{\mathcal{R}(\mathbf{w})}{\epsilon}$$

for $t \in \{\tau, \ldots, l \cdot \Gamma\}$, and:

$${\boldsymbol{\pi}^1}_t(s) = \arg\max_{\mathbf{w} \in \mathcal{P}_{A_1}} \frac{\mathcal{R}(\mathbf{w})}{\epsilon} \text{ for } t = \tau.$$

$\mathbf{Q}_k(s)$ denotes a row of matrix $\mathbf{Q}_k$ (see Section 2.3),[4] $\mathbf{w}$ is a row vector from probability simplex $\mathcal{P}_{A_1}$, $\dagger$ denotes the transpose operator, $\mathcal{R}$ is a 1-strongly convex regularizer w.r.t. norm $\|\cdot\|_1$, and ϵ is the learning rate. This approach, henceforth referred to as *experts with periodic restarts* (EXPRESTART), suffices to obtain sublinear regret provided that the segment length Γ and learning rate ϵ are properly set (see Appendix G of Radanovic et al. (2019)).

One of the main drawbacks of experts with periodic restarts is that it potentially results in abrupt changes in the policy of agent $\mathcal{A}_1$, this occurring when switching from one segment to another. In practice, one might want to avoid this, for example, because agent $\mathcal{A}_2$ (e.g., representing a person) might negatively respond to such abrupt changes in agent $\mathcal{A}_1$'s policy. Considering this, we design a new experts algorithm that ensures gradual policy changes for agent $\mathcal{A}_1$ across episodes, while achieving the same order of regret guarantees (see Section 5.4 and Appendix G of Radanovic et al. (2019)).

4.2. Experts with Double Recency Bias: EXPDRBIAS

Utilizing fixed segments, as in the approach of EXPRESTART, leads to potentially rapid policy changes after each segment. To avoid this issue, we can for each episode t consider a family of segments of different lengths: $\{t-\Gamma, ..., t\}$, $\{t-\Gamma+1, ..., t\}$, ..., $\{t-1, t\}$, and run the OFTRL algorithm on each segment separately. The policy in episode t can then be defined as the average of the OFTRL outputs. This approach, henceforth referred to as *experts with double recency bias* (EXPDRBIAS), can be implemented through the following two ideas that bias the policy selection rule towards recent information in a twofold manner.

Recency Windowing. The first idea is what we call *recency windowing*. Simply put, it specifies how far in the history an agent should look when choosing a policy. More precisely, we define a sliding window of size Γ and to decide on policy ${\boldsymbol{\pi}^1}_t$ we only use historical information from periods after $t-\Gamma$. In particular, the updating rule of OFTRL would be modified for $t>1$ as ${\boldsymbol{\pi}^1}_t(s) =$

$$\arg\max_{\mathbf{w}\in\mathcal{P}_{A_1}} \left(\sum_{k=\max(1,t-\Gamma)}^{t-1} \mathbf{Q}_k(s) + \mathbf{Q}_{t-1}(s) \right) \mathbf{w}^\dagger + \frac{\mathcal{R}(\mathbf{w})}{\epsilon}.$$

and:

$${\boldsymbol{\pi}^1}_1(s) = \arg\max_{\mathbf{w}\in\mathcal{P}_{A_1}} \frac{\mathcal{R}(\mathbf{w})}{\epsilon} \text{ for } t=1. \tag{2}$$

Recency Modulation. The second idea is what we call *recency modulation*. This creates an averaging effect over the policies computed by the experts with periodic restarts approach, for different possible starting points of the segmentation. For episode t, recency modulation calculates policy updates using recency windowing but considers windows of different sizes. More precisely, we calculate updates with window sizes 1 to Γ, and then average them to obtain the final update. Lemma 3 shows that this updating rule will not lead to abrupt changes in agent $\mathcal{A}_1$'s policy.

To summarize, agent $\mathcal{A}_1$ has the following policy update rule for $t>1$:

$${\boldsymbol{\pi}^1}_t(s) = \frac{1}{\Gamma} \sum_{\tau=1}^{\Gamma} \mathbf{w}_{t,\tau}(s), \tag{3}$$

where $\mathbf{w}_{t,\tau}(s) =$

$$\arg\max_{\mathbf{w}\in\mathcal{P}_{A_1}} \left(\sum_{k=\max(1,t-\tau)}^{t-1} \mathbf{Q}_k(s) + \mathbf{Q}_{t-1}(s) \right) \mathbf{w}^\dagger + \frac{\mathcal{R}(\mathbf{w})}{\epsilon}.$$

For $t=1$, we follow equation update (2). The full description of agent $\mathcal{A}_1$'s policy update using the approach of EXPDRBIAS is given in Algorithm 1. As with EXPRESTART, EXPDRBIAS leads to a sub-linear regret for a proper choice of ϵ and Γ, which in turn results in a near-optimal behavior, as analyzed in the next section.

Algorithm 1: EXPDRBIAS

Input: History horizon Γ, learning rate ϵ
begin
  Initialize: $\forall s$, compute ${\boldsymbol{\pi}^1}_1(s)$ using Eq. (2)
  for *episode* $t \in \{1, ..., T\}$ **do**
    $\forall s$, commit to policy ${\boldsymbol{\pi}^1}_t(s)$
    Obtain the return V_t
    Observe agent $\mathcal{A}_2$'s policy ${\boldsymbol{\pi}^2}_t$
    Calculate Q-values $\mathbf{Q}_t$
    $\forall s$, compute ${\boldsymbol{\pi}^1}_{t+1}(s)$ using Eq. (3)
  end
end

5. Theoretical Analysis of EXPDRBIAS

To bound regret $R(T)$, given by equation (1), it is useful to express difference $\eta_t(\boldsymbol{\pi}^1) - \eta_t({\boldsymbol{\pi}^1}_t)$ in terms of Q-values. In particular, one can show that this difference is equal to $\boldsymbol{d}_{\boldsymbol{\pi}^1,{\boldsymbol{\pi}^2}_t} \cdot (\mathbf{Q}_t{}^{\boldsymbol{\pi}^1} - \mathbf{Q}_t{}^{{\boldsymbol{\pi}^1}_t})$ (see Lemma 15 in Appendix B of Radanovic et al. (2019)). By the definitions of $\mathbf{Q}_t{}^{\boldsymbol{\pi}^1}$ and $\mathbf{Q}_t{}^{{\boldsymbol{\pi}^1}_t}$, this implies:

$$\eta_t(\boldsymbol{\pi}^1) - \eta_t({\boldsymbol{\pi}^1}_t) = \boldsymbol{d}_{\boldsymbol{\pi}^1,{\boldsymbol{\pi}^2}_t} \cdot \left\langle \boldsymbol{\pi}^1 - {\boldsymbol{\pi}^1}_t, \mathbf{Q}_t \right\rangle.$$

If $\boldsymbol{d}_{\boldsymbol{\pi}^1,{\boldsymbol{\pi}^2}_t}$ was not dependent on t (e.g., if agent $\mathcal{A}_2$ was not changing its policy), then bounding $R(T)$ would amount to

[4] Given ${\boldsymbol{\pi}^1}_t$ and ${\boldsymbol{\pi}^2}_t$, we can calculate $\mathbf{Q}_t$ from the Bellman equation using standard dynamic programming techniques.

bounding the sum of terms $\langle \boldsymbol{\pi}^1 - \boldsymbol{\pi}^1{}_t, \mathbf{Q}_t \rangle$. This could be done with an approach that carefully combines the proof techniques of Even-Dar et al. (2005) with the OFTRL properties, in particular, *regret bounded by variation in utilities* (RVU) (Syrgkanis et al., 2015). However, in our setting $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2{}_t}$ is generally changing with t.

5.1. Change Magnitudes of Stationary Distributions

To account for this, we need to investigate how quickly distributions $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2{}_t}$ change across episodes. Furthermore, to utilize the RVU property, we need to do the same for distributions $\boldsymbol{d}_{\boldsymbol{\pi}^1{}_t, \boldsymbol{\pi}^2{}_t}$. The following lemma provides bounds on the respective change magnitudes.

Lemma 2. *The difference between the stationary distributions of two consecutive episodes is upper bounded by:*

$$\left\| \boldsymbol{d}_{\boldsymbol{\pi}^1{}_t, \boldsymbol{\pi}^2{}_t} - \boldsymbol{d}_{\boldsymbol{\pi}^1{}_{t-1}, \boldsymbol{\pi}^2{}_{t-1}} \right\|_1 \leq \frac{\rho_1 + I_2 \cdot \rho_2}{1 - e^{-\frac{1}{\omega}}}.$$

Furthermore, for any policy $\boldsymbol{\pi}^1$:

$$\left\| \boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2{}_t} - \boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2{}_{t-1}} \right\|_1 \leq \frac{I_2 \cdot \rho_2}{1 - e^{-\frac{1}{\omega}}}.$$

5.2. Properties Based on OFTRL

The bounds on the change magnitudes of distributions $\boldsymbol{d}_{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2{}_t}$ and $\boldsymbol{d}_{\boldsymbol{\pi}^1{}_t, \boldsymbol{\pi}^2{}_t}$, which will propagate to the final result, depend on agent $\mathcal{A}_1$'s policy change magnitude ρ_1. The following lemma provides a bound for ρ_1 that, together with the assumed bound on ρ_2, is useful in establishing no-regret guarantees.

Lemma 3. *For any $t > 1$ and $1 < \tau \leq \Gamma$, the change magnitude of weights $\mathbf{w}_{t,\tau}$ in* ExpDRBias *is bounded by:*

$$\left\| \mathbf{w}_{t,\tau} - \mathbf{w}_{t-1,\tau-1} \right\|_\infty \leq \min \left\{ 2, \frac{9 \cdot \epsilon}{1 - e^{-\frac{1}{\omega}}} \right\}$$

Consequently:

$$\rho_1 \leq \min \left\{ 2, \frac{9 \cdot \epsilon}{1 - e^{-\frac{1}{\omega}}} + \frac{2}{\Gamma} \right\}.$$

Now, we turn to bounding the term $\langle \boldsymbol{\pi}^1 - \boldsymbol{\pi}^1{}_t, \mathbf{Q}_t \rangle$. Lemma 4 formalizes the RVU property for ExpDRBias using the L_1 norm and its dual L_∞ norm, derived from results in the existing literature (Syrgkanis et al., 2015).[5] Lemma 4 shows that it is possible to bound $\langle \boldsymbol{\pi}^1 - \boldsymbol{\pi}^1{}_t, \mathbf{Q}_t \rangle$ by examining the change magnitudes of Q-values.

[5] An extended version of the lemma, which is needed for the main result, is provided in Appendix E.1 of Radanovic et al. (2019).

Lemma 4. *Consider* ExpDRBias *and let* $\mathbf{1}$ *denote column vector of ones with $|S|$ elements. Then, for each episode $1 \leq t \leq T - \Gamma + 1$ of* ExpDRBias *, we have:*

$$\begin{aligned}
&\sum_{\tau=1}^{\Gamma} \langle \boldsymbol{\pi}^1 - \mathbf{w}_{t+\tau-1,\tau}, \mathbf{Q}_{t+\tau-1} \rangle \\
&\quad \leq \mathbf{1} \cdot \left(\frac{\Delta_{\mathcal{R}}}{\epsilon} + \epsilon \cdot \sum_{\tau=1}^{\Gamma} \left\| \mathbf{Q}_{t+\tau-1} - \mathbf{Q}_{t+\tau-2} \right\|_{\max}^2 \right. \\
&\quad \left. - \frac{1}{4 \cdot \epsilon} \cdot \sum_{\tau=1}^{\Gamma} \left\| \mathbf{w}_{t+\tau-1,\tau} - \mathbf{w}_{t+\tau-2,\tau-1} \right\|_\infty^2 \right),
\end{aligned}$$

where $\mathbf{w}_{t+\tau-1,\tau}$ are defined in (3), $\Delta_{\mathcal{R}} = \sup_{\mathbf{w} \in \mathcal{P}_{A_1}} \mathcal{R}(\mathbf{w}) - \inf_{\mathbf{w} \in \mathcal{P}_{A_1}} \mathcal{R}(\mathbf{w})$, and $\boldsymbol{\pi}^1$ is an arbitrary policy of agent $\mathcal{A}_1$.

5.3. Change Magnitudes of Q-values

We now derive bounds on the change magnitudes of Q-values that we use together with Lemma 4 to prove the main results. We first bound the difference $\mathbf{Q}_t^{\boldsymbol{\pi}^1{}_t} - \mathbf{Q}_{t-1}^{\boldsymbol{\pi}^1{}_{t-1}}$, which helps us in bounding the difference $\mathbf{Q}_t - \mathbf{Q}_{t-1}$.

Lemma 5. *The difference between $\mathbf{Q}_t^{\boldsymbol{\pi}^1{}_t}$-values of two consecutive episodes is upper bounded by:*

$$\left\| \mathbf{Q}_t^{\boldsymbol{\pi}^1{}_t} - \mathbf{Q}_{t-1}^{\boldsymbol{\pi}^1{}_{t-1}} \right\|_\infty \leq C_{Q^{\boldsymbol{\pi}}},$$

where $C_{Q^{\boldsymbol{\pi}}} = 3 \cdot \frac{\rho_1 + I_2 \cdot \rho_2}{(1 - e^{-\frac{1}{\omega}})^2} + 2 \cdot \frac{\rho_1 + \rho_2}{1 - e^{-\frac{1}{\omega}}}$.

Lemma 6. *The difference between $\mathbf{Q}_t$-values of two consecutive episodes is upper bounded by:*

$$\left\| \mathbf{Q}_t - \mathbf{Q}_{t-1} \right\|_{\max}^2 \leq C_Q^2,$$

where $C_Q = C_{Q^{\boldsymbol{\pi}}} + \left(\frac{3}{1 - e^{\frac{1}{\omega}}} + 1 \right) \cdot \rho_1 + 2 \cdot \rho_2 + \frac{\rho_1 + I_2 \cdot \rho_2}{1 - e^{-\frac{1}{\omega}}}$.

For convenience, instead of directly using C_Q, we consider a variable C_ω such that $C_\omega \geq \frac{C_Q}{\max\{\rho_1, \rho_2\}}$. The following proposition gives a rather loose (but easy to interpret) bound on C_ω that satisfies the inequality.

Proposition 1. *There exists a constant C_ω independent of ρ_1 and ρ_2, such that:*[6]

$$\frac{C_Q}{\max\{\rho_1, \rho_2\}} \leq C_\omega \leq \frac{18}{(1 - e^{-\frac{1}{\omega}})^2}$$

Proof. The claim is directly obtained from Lemma 5, Lemma 6, and the fact that $I_2 \leq 1$ and $\frac{1}{1 - e^{-\frac{1}{\omega}}} \geq 1$. □

[6] When $\rho_1 = \rho_2 = 0$, $C_\omega \geq 1$.

5.4. Regret Analysis and Main Results

We now come to the most important part of our analysis: establishing the regret guarantees for ExpDRBias. Using the results from the previous subsections, we obtain the following regret bound:

Theorem 1. *Let the learning rate of* ExpDRBias *be equal to* $\epsilon = \frac{1}{\Gamma^{\frac{1}{4}}}$ *and let* $k > 0$ *be such that* $\rho_1 \leq k \cdot \epsilon$ *and* $\rho_2 \leq k \cdot \epsilon$. *Then, the regret of* ExpDRBias *is upper-bounded by:*

$$R(T) \leq 2 \cdot \left(\Delta_{\mathcal{R}} + k^2 \cdot C_\omega^2\right) \cdot T \cdot \Gamma^{-\frac{3}{4}} + \frac{6 \cdot I_2 \cdot \rho_2}{(1 - e^{-\frac{1}{\omega}})^2} \cdot T \cdot \Gamma.$$

When agent $\mathcal{A}_2$ does not influence the transition kernel through its policy, i.e., when $I_2 = 0$, the regret is $\mathcal{O}(T^{\frac{1}{4}})$ for $\Gamma = T$. In this case, we could have also applied the original approach of Even-Dar et al. (2005; 2009), but interestingly, it would result in a worse regret bound, i.e., $\mathcal{O}(T^{\frac{1}{2}})$. By leveraging the fact that agent $\mathcal{A}_2$'s policy is slowly changing, which corresponds to reward functions in the setting of Even-Dar et al. (2005; 2009) not being fully adversarial, we are able to improve on the worst-case guarantees. The main reason for such an improvement is our choice of the underlying experts algorithm, i.e., OFTRL, that exploits the apparent predictability of agent $\mathcal{A}_2$'s behavior. Similar arguments were made for the repeated games settings (Rakhlin & Sridharan, 2013; Syrgkanis et al., 2015), which correspond to our setting when the MDP consists of only one state. Namely, in the single state scenario, agent $\mathcal{A}_2$ does not influence transitions, so the resulting regret is $\mathcal{O}(T^{\frac{1}{4}})$, matching the results of Syrgkanis et al. (2015).

In general, the regret depends on ρ_2. If $\rho_2 = \mathcal{O}(T^{-\alpha})$ with $0 < \alpha \leq \frac{7}{4}$, then $\Gamma = \mathcal{O}(T^{\frac{4}{7} \cdot \alpha})$ equalizes the order of the two regret components in Theorem 1 and leads to the regret of $\mathcal{O}(T^{1 - \frac{3}{7} \cdot \alpha})$. This brings us to the main result, which provides a lower bound on the return $\bar{V}$:

Theorem 2. *Assume that* $\rho_2 = \mathcal{O}(T^{-\alpha})$ *for* $\alpha > 0$. *Let* $\epsilon = \frac{1}{\Gamma^{\frac{1}{4}}}$ *and* $\Gamma = \min\{T^{\frac{4}{7} \cdot \alpha}, T\}$. *Then, the regret of* ExpDRBias *is upper-bounded by:*

$$R(T) = \mathcal{O}(T^{\max\{1 - \frac{3}{7} \cdot \alpha, \frac{1}{4}\}}).$$

Furthermore, when the MDP is (λ, μ)-smooth*, the return of* ExpDRBias *is lower-bounded by:*

$$\bar{V} \geq \frac{\lambda}{1 + \mu} \cdot \textsc{Opt} - \mathcal{O}(T^{\max\{-\frac{3}{7} \cdot \alpha, -\frac{3}{4}\}}) - \mathcal{O}(M^{-1}).$$

Proof. Notice that $\frac{9 \cdot \epsilon}{1 - e^{-\frac{1}{\omega}}} \geq \frac{2}{\Gamma}$ for $\Gamma \geq 1$. By Lemma 3, this implies that there exists a fixed k (not dependant on T) such that $\rho_1 \leq k \cdot \epsilon$ for large enough T. Furthermore, $\epsilon = T^{-\min\{\frac{4}{7} \cdot \alpha, \frac{1}{4}\}}$, so there exists a fixed k such that $\rho_2 \leq k \cdot \epsilon$ for large enough T. Hence, we can apply Theorem 1 to obtain an order-wise regret bound: $\mathcal{O}(T \cdot \Gamma^{-\frac{3}{4}}) + \mathcal{O}(\rho_2 \cdot T \cdot \Gamma)$.

Now, consider two cases. First, let $\alpha \leq \frac{7}{4}$. Then, we obtain:

$$\begin{aligned} R(T) &= \mathcal{O}(T \cdot \Gamma^{-\frac{3}{4}}) + \mathcal{O}(\rho_2 \cdot T \cdot \Gamma) \\ &= \mathcal{O}(T \cdot T^{-\frac{3}{4} \cdot \frac{4}{7} \cdot \alpha}) + \mathcal{O}(T^{-\alpha} \cdot T \cdot T^{\frac{4}{7} \cdot \alpha}) = \mathcal{O}(T^{1 - \frac{3}{7} \cdot \alpha}). \end{aligned}$$

For the other case, i.e., when $\alpha \geq \frac{7}{4}$, we obtain:

$$\begin{aligned} R(T) &= \mathcal{O}(T \cdot \Gamma^{-\frac{3}{4}}) + \mathcal{O}(\rho_2 \cdot T \cdot \Gamma) \\ &= \mathcal{O}(T \cdot T^{-\frac{3}{4}}) + \mathcal{O}(T^{-\alpha} \cdot T \cdot T) = \mathcal{O}(T^{\frac{1}{4}}). \end{aligned}$$

Therefore, $R(T) = \mathcal{O}(T^{\max\{1 - \frac{3}{7} \cdot \alpha, \frac{1}{4}\}})$, which proves the first statement. By combining it with Lemma 1, we obtain the second statement. □

The multiplicative factors in the asymptotic bounds mainly depend on mixing time ω. In particular they are dominated by factor $\frac{1}{1 - e^{-\frac{1}{\omega}}}$ and its powers, as can be seen from Lemma 1, Theorem 1, and Proposition 1. Note that Lemma 3 allows us to upper bound k in Theorem 1 with $\mathcal{O}\left(\frac{1}{1 - e^{-\frac{1}{\omega}}}\right)$. Furthermore, $\frac{1}{1 - e^{-\frac{1}{\omega}}} \approx \omega$ for large enough ω. Hence, these results imply $\mathcal{O}(\omega^6)$ dependency of the asymptotic bounds on ω. This is larger than what one might expect from the prior work, for example the bound in Even-Dar et al. (2005; 2009) has $\mathcal{O}(\omega^2)$ dependency. However, our setting is different, in that the presence of agent $\mathcal{A}_2$ has an effect on transitions (from agent $\mathcal{A}_1$'s perspective), and so it is not surprising that the resulting dependency on the mixing time is worse.

6. Hardness Result

Our formal guarantees assume that the policy change magnitude ρ_2 of agent $\mathcal{A}_2$ is a decreasing function in the number of episodes given by $\mathcal{O}(T^{-\alpha})$ for $\alpha > 0$. What if we relax this, and allow agent $\mathcal{A}_2$ to adapt independently of the number of episodes? We show a hardness result for the setting of $\alpha = 0$, using a reduction from the *online agnostic parity learning* problem (Abbasi et al., 2013). As argued in Abbasi et al. (2013), the online to batch reduction implies that the online version of agnostic parity learning is at least as hard as its offline version, for which the best known algorithm has complexity $2^{\mathcal{O}(\frac{n}{\log n})}$ (Kalai et al., 2008). In fact, agnostic parity learning is a harder variant of the *learning with parity noise* problem, widely believed to be computationally intractable (Blum et al., 2003; Pietrzak, 2012), and thus often adopted as a hardness assumption (e.g., Sharan et al. (2018)).

Theorem 3. *Assume that the policy change magnitude ρ_2 of agent $\mathcal{A}_2$ is order $\Omega(1)$ and that its influence is $I_2 = 1$. If there exists a $poly(|S|, T)$ time algorithm that outputs a policy sequence ${\boldsymbol{\pi}^1}_1$, ..., ${\boldsymbol{\pi}^1}_T$ whose regret is $\mathcal{O}(poly(|S|) \cdot T^\beta)$ for $\beta < 1$, then there also exists a $poly(|S|, T)$ time algorithm for online agnostic parity learning whose regret is $\mathcal{O}(poly(|S|) \cdot T^\beta)$.*

The proof relies on the result of Abbasi et al. (2013) (Theorem 5), which reduces the *online agnostic parity learning* problem to the *adversarial shortest path* problem, which we reduce to our problem. Theorem 3 implies that when $\alpha = 0$, it is unlikely to obtain $R(T)$ that is sub-linear in T given the current computational complexity results.

7. Related Work

Experts Learning in MDPs. Our framework is closely related to that of Even-Dar et al. (2005; 2009), although the presence of agent $\mathcal{A}_2$ means that we cannot directly use their algorithmic approach. In fact, learning with an arbitrarily changing transition is believed to be computationally intractable (Abbasi et al., 2013), and computationally efficient learning algorithms experience linear regret (Yu & Mannor, 2009a; Abbasi et al., 2013). This is where we make use of the bound on the magnitude of agent $\mathcal{A}_2$'s policy change. Contrary to most of the existing work, the changes in reward and transition kernel in our model are non-oblivious and adapting to the learning algorithm of agent $\mathcal{A}_1$. There have been a number of follow-up works that either extend these results or improve them for more specialized settings (Dick et al., 2014; Neu et al., 2012; 2010; Dekel & Hazan, 2013). Agarwal et al. (2017) and Singla et al. (2018) study the problem of learning with experts advice where experts are not stationary and are learning agents themselves. However, their focus is on designing a meta-algorithm on how to coordinate with these experts and is technically very different from ours.

Learning in Games. To relate the quality of an optimal solution to agent $\mathcal{A}_1$'s regret, we use techniques similar to those studied in the learning in games literature (Blum et al., 2008; Roughgarden, 2009; Syrgkanis et al., 2015). The fact that agent $\mathcal{A}_2$'s policy is changing slowly enables us to utilize no-regret algorithms for learning in games with recency bias (Daskalakis et al., 2011; Rakhlin & Sridharan, 2013; Syrgkanis et al., 2015), providing better regret bounds than through standard no-regret learning techniques (Littlestone & Warmuth, 1994; Freund & Schapire, 1997). The recent work by Wei et al. (2017) studies two-player learning in zero-sum stochastic games. Apart from focusing on zero-sum games, Wei et al. (2017) adopt a different set of assumptions to derive regret bounds and their results are not directly comparable to ours. Furthermore, their algorithmic techniques are orthogonal to those that we pursue; these differences are elaborated in Wei et al. (2017).

Human AI Collaboration. The helper-AI problem (Dimitrakakis et al., 2017) is related to the present work, in that an AI agent is designing its policy by accounting for human imperfections. The authors use a Stackleberg formulation of the problem in a single shot scenario. Their model assumes that the AI agent knows the behavioral model of the human agent, which is a best response to the policy of the AI agent for an incorrect transition kernel. We relax this requirement by studying a repeated human-AI interaction. Nikolaidis et al. (2017) study a repeated human-AI interaction, but their setting is more restrictive than ours as they do not model the changes in the environment. In particular, they have a repeated game setup, where the only aspect that changes over time is the "state" of the human representing what knowledge the human has about the robot's payoffs. Prior work also considers a learner that is aware of the presence of other actors (Foerster et al., 2018; Raileanu et al., 2018). While these multi-agent learning approaches account for the evolving behavior of other actors, the underlying assumption is typically that each agent follows a known model.

Steering and Teaching. There is also a related literature on "steering" the behavior of other agent. For example, (i) the *environment design* framework of Zhang et al. (2009), where one agent tries to steer the behavior of another agent by modifying its reward function, (ii) the *cooperative inverse reinforcement learning* of Hadfield-Menell et al. (2016), where the human uses demonstrations to reveal a proper reward function to the AI agent, and (iii) the *advice-based interaction* model (Amir et al., 2016), where the goal is to communicate advice to a sub-optimal agent on how to act in the world. The latter approach is also in close relationship to the *machine teaching* literature (Zhu et al., 2018; Zhu, 2015; Singla et al., 2013; Cakmak & Lopes, 2012). Our work differs from this literature; we focus on joint decision-making, rather than teaching or steering.

8. Conclusion

In this paper, we have presented a two-agent MDP framework in a collaborative setting. We considered the problem of designing a no-regret algorithm for the first agent in the presence of an adapting, second agent (for which we make no assumptions about its behavior other than a requirement that it adapts slowly enough). Our algorithm builds from the ideas of experts learning in MDPs, and makes use of a novel form of recency bias to achieve strong regret bounds. In particular, we showed that in order for the first agent to facilitate collaboration, it is critical that the second agent's policy changes are not abrupt. An interesting direction for future work would be to consider the partial information setting, in which, for example, agent $\mathcal{A}_1$ has only a noisy estimate of agent $\mathcal{A}_2$'s policy.

Acknowledgements

This work was supported in part by a SNSF Early Postdoc Mobility fellowship.

References

Abbasi, Y., Bartlett, P. L., Kanade, V., Seldin, Y., and Szepesvári, C. Online learning in markov decision processes with adversarially chosen transition probability distributions. In *NIPS*, pp. 2508–2516, 2013.

Agarwal, A., Luo, H., Neyshabur, B., and Schapire, R. E. Corralling a band of bandit algorithms. In *COLT*, pp. 12–38, 2017.

Amir, O., Kamar, E., Kolobov, A., and Grosz, B. Interactive teaching strategies for agent training. In *IJCAI*, pp. 804–811, 2016.

Blum, A., Kalai, A., and Wasserman, H. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

Blum, A., Hajiaghayi, M., Ligett, K., and Roth, A. Regret minimization and the price of total anarchy. In *STOC*, pp. 373–382. ACM, 2008.

Boutilier, C. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pp. 195–210, 1996.

Cakmak, M. and Lopes, M. Algorithmic and human teaching of sequential decision tasks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Daskalakis, C., Deckelbaum, A., and Kim, A. Near-optimal no-regret algorithms for zero-sum games. In *SODA*, pp. 235–254, 2011.

Dekel, O. and Hazan, E. Better rates for any adversarial deterministic mdp. In *ICML*, pp. 675–683, 2013.

Dick, T., Gyorgy, A., and Szepesvari, C. Online learning in markov decision processes with changing cost sequences. In *ICML*, pp. 512–520, 2014.

Dimitrakakis, C., Parkes, D. C., Radanovic, G., and Tylkin, P. Multi-view decision processes: The helper-ai problem. In *NIPS*, pp. 5443–5452, 2017.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Experts in a markov decision process. In *NIPS*, pp. 401–408, 2005.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. In *AAMAS*, pp. 122–130, 2018.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1): 119–139, 1997.

Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. In *NIPS*, pp. 3909–3917, 2016.

Kalai, A. T., Mansour, Y., and Verbin, E. On agnostic boosting and parity learning. In *STOC*, pp. 629–638, 2008.

Kanade, V. and Steinke, T. Learning hurdles for sleeping experts. *ACM Transactions on Computation Theory (TOCT)*, 6(3):11, 2014.

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

Neu, G., Antos, A., György, A., and Szepesvári, C. Online markov decision processes under bandit feedback. In *NIPS*, pp. 1804–1812, 2010.

Neu, G., Gyorgy, A., and Szepesvári, C. The adversarial stochastic shortest path problem with unknown transition probabilities. In *AISTATS*, pp. 805–813, 2012.

Nikolaidis, S., Nath, S., Procaccia, A. D., and Srinivasa, S. Game-theoretic modeling of human adaptation in human-robot collaboration. In *Proceedings of the International conference on human-robot interaction*, pp. 323–331, 2017.

Pietrzak, K. Cryptography from learning parity with noise. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 99–114, 2012.

Radanovic, G., Devidze, R., Parkes, D., and Singla, A. Learning to collaborate in markov decision processes. *arXiv preprint arXiv:1901.08029*, 2019.

Raileanu, R., Denton, E., Szlam, A., and Fergus, R. Modeling others using oneself in multi-agent reinforcement learning. In *ICML*, pp. 4254–4263, 2018.

Rakhlin, S. and Sridharan, K. Optimization, learning, and games with predictable sequences. In *NIPS*, 2013.

Roughgarden, T. Intrinsic robustness of the price of anarchy. In *STOC*, pp. 513–522. ACM, 2009.

Sharan, V., Kakade, S., Liang, P., and Valiant, G. Prediction with a short memory. In *STOC*, pp. 1074–1087, 2018.

Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., and Krause, A. On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*, 2013.

Singla, A., Hassani, S. H., and Krause, A. Learning to interact with learning agents. In *AAAI*, 2018.

Syrgkanis, V., Agarwal, A., Luo, H., and Schapire, R. E. Fast convergence of regularized learning in games. In *NIPS*, pp. 2989–2997, 2015.

Wei, C.-Y., Hong, Y.-T., and Lu, C.-J. Online reinforcement learning in stochastic games. In *NIPS*, pp. 4987–4997, 2017.

Yu, J. Y. and Mannor, S. Arbitrarily modulated markov decision processes. In *Decision and Control, 2009*, pp. 2946–2953, 2009a.

Yu, J. Y. and Mannor, S. Online learning in markov decision processes with arbitrarily changing rewards and transitions. In *GameNets*, pp. 314–322, 2009b.

Yu, J. Y., Mannor, S., and Shimkin, N. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Zhang, H., Parkes, D. C., and Chen, Y. Policy teaching through reward function learning. In *EC*, pp. 295–304, 2009.

Zhu, X. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pp. 4083–4087, 2015.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.