Provably Efficient Model-free RL in Leader-Follower MDP with Linear Function Approximation

Arnob Ghosh GHOSH.244@OSU.EDU
Electrical and Computer Engg., The Ohio State University

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

We consider a multi-agent episodic MDP setup where an agent (leader) takes action at each step of the episode followed by another agent (follower). The state evolution and rewards depend on the joint action pair of the leader and the follower. Such types of interactions can find applications in many domains such as smart grids, mechanism design, security, and policymaking. We are interested in how to learn policies for both the players with provable performance guarantee under a bandit feedback setting. We focus on a setup where both the leader and followers are *non-myopic*, i.e., they both seek to maximize their rewards over the entire episode and consider a linear MDP which can model continuous state-space which is very common in many RL applications. We propose a *model-free* RL algorithm and show that $\tilde{O}(\sqrt{d^3H^3T})$ regret bounds can be achieved for both the leader and the follower, where d is the dimension of the feature mapping, H is the length of the episode, and T is the total number of steps under the bandit feedback information setup. *Thus, our result holds even when the number of states becomes infinite*. The algorithm relies on *novel* adaptation of the single agent LSVI-UCB algorithm. Specifically, we replace the standard greedy policy (as the best response) with the soft-max policy for both the leader and the follower. This turns out to be key in establishing uniform concentration bound for the value functions. To the best of our knowledge, this is the first sub-linear regret bound guarantee for the Markov games with non-myopic followers with function approximation.

1. Introduction

Multi-agent Reinforcement Learning (MARL) has become an important tool for decision-making in a Markov game. In many sequential real-world decision-making problems, agents often have asymmetric roles. For example, one agent (leader) can act first and observe the action of the leader, and the other agent (follower) reacts at each step of the MDP. This type of interaction requires two levels of thinking: the leader must reason what the follower would do in order to find its optimal decision. For example, an electric utility company (the *leader*) seeks to maximize social welfare by selecting prices at different times over a day while the users (the *followers*) seek to optimize their own consumption based on the prices set by the utility company. The reward and the underlying transition probability depend on both the leader's and the follower's actions. Such leader-follower interactions appear in other applications as well such as in AI Economist (Zheng et al., 2020), Mechanism Design (Conitzer and Sandholm, 2004), optimal auction (Cole and Roughgarden, 2014), and security games (Tambe, 2011).

Such kind of leader-follower interaction is different from the simultaneous play in the Markov setting as considered in Jin et al. (2021); Tian et al. (2021). In general, the Stackelberg equilibrium is the relevant concept for this type of leader-follower interaction (Conitzer and Sandholm, 2006) compared to the Nash equilibrium considered in the above paper. Letchford et al. (2009); Peng et al. (2019) considered a learning framework in order to learn the Stackelberg equilibrium with a best-response oracle for the follower. However, these works can not be generalized to the *bandit feedback* setting where the leader and the follower can only observe those rewards corresponding to the state-actions pairs encountered. Efficient learning in this leader-follower MDP setting under the bandit feedback (which is more natural) is fundamentally more challenging compared to the single-agent setting due to the more challenging exploration-exploitation trade-off.

Few recent works have focused on such kind of leader-follower interaction in MDP with the bandit feedback setting. Bai et al. (2021) considered a model where the leader selects the underlying MDP on which the follower acts on, i.e., the leader only acts once at the start of the episode. However, we consider the setup where both the leader and the follower interact at every step of the episode. Kao et al. (2022) considers a setup where both the leader and the follower interact at every step similar to ours. However, Kao et al. (2022) considered the setup where both the leader and the follower receive the same reward whereas in our setting the rewards can be different. Further, both the above papers consider the finite state-space (a.k.a. tabular setup) where the sample complexity scales with the state space. Thus, the above approach *would not be useful* for large-scale RL applications where the number of states could even be infinite. To address this curse of dimensionality, modern RL has adopted *function approximation* techniques to approximate the (action-) value function or a policy, which greatly expands the potential reach of RL, especially via deep neural networks. For large state-space the model-based approaches as considered in the above papers have limited application (Wei et al., 2020), thus, we focus on developing a model-free algorithm. Only Zhong et al. (2021) considers a leader-follower Markov setup with function approximation. However, they consider *myopic* followers who seek to maximize instantaneous reward and also consider the followers' rewards are known. Hence, it greatly alleviates the exploration challenge as it is only limited to the leader's side. Rather, we consider the setup where the followers are also non-myopic with bandit feedback. Thus, we seek to answer the following question

Can we achieve provably optimal regret for model-free exploration for leader-follower (non-myopic) interaction in MDP with function approximation under bandit feedback?

Our Contribution: To answer the above question, we consider the leader-follower Markov game with linear function approximation (bandit feedback) where at each step of the sequential decision process the leader takes an action and observes the action the follower reacts. The transition probability and the reward functions can be represented as a *linear function* of some known feature mapping adapted from the single agent set up in Jin et al. (2020). Our main contributions are:

- We show that with a proper parameter choice, our proposed model-free algorithm achieves $\tilde{\mathcal{O}}(\sqrt{d^3H^3T})$ regret for both leader and follower, where d is the dimension of the feature mapping, H is the length of the episode, and T is the total number of steps. Note that for the single agent setup, the regret is of the same order achieved in Jin et al. (2020). Hence, our result matches the regret bound for the single-agent setup. Our regret bounds also enable us to obtain $\tilde{\mathcal{O}}(\sqrt{d^3H^4/K})$-Coarse Correlated Stackelberg equilibrium policy.

- Our bounds are attained without explicitly estimating the unknown transition model or *requiring any simulator or best-response oracle*, and they depend on the state space only through the dimension of the feature mapping. *To the best of our knowledge, these sub-linear regret bounds are the first results for the leader-follower non-myopic MDP game with function approximations under bandit feedback.* Since linear MDP contains a tabular setup, as a by-product, we provide the first result on the regret bound for the leader-follower (non-myopic) MDP game under bandit feedback using the model-free RL algorithm.

- We adapt the classic model-free LSVI-UCB algorithm proposed in Jin et al. (2020) in a novel manner. Due to the nature of the leader-follower interaction, a key challenge arises while establishing the value-aware uniform concentration, which lies at the heart of the performance analysis of model-free exploration. In particular, for a single agent set-up, the greedy selection with respect to the standard Q-function achieves a small covering number. However, in the game setting, for a given policy, the best response strategy of a player fails to achieve such a non-trivial covering number for the value-function class of the players (i.e., V-function). To address this fundamental issue, we instead adopt a soft-max policy for the players by utilizing its nice property of approximation-smoothness trade-off via its parameter, i.e., temperature coefficient.

Related Literature: Provably efficient RL algorithms for zero-sum Markov games have been proposed (Wei et al., 2017; Bai et al., 2020; Liu et al., 2021; Xie et al., 2020; Chen et al., 2021; Sayin et al., 2021). Provably efficient algorithms to obtain coarse correlated equilibrium have also been proposed for the general sum-game as well (Bai and Jin, 2020; Jin et al., 2021; Mao and Başar, 2022). In contrast to the above papers, we consider a leader-follower setup. Hence, our work is not directly comparable.

For learning Stackelberg equilibrium, most of the works focus on the normal form game which is equivalent to step size $H = 1$ in our setting (Balcan et al., 2015; Blum et al., 2014; Peng et al., 2019; Letchford et al., 2009). Further, in the above papers, it is assumed that the followers' responses are known, in contrast, in our setting the follower is also learning its optimal policy. Zhong et al. (2021) considered a leader-follower setup with myopic follower and known reward for linear MDP which alleviates the challenges of exploration for the follower. Zhong et al. (2021) also proposed a model-based approach for tabular setup with myopic followers and unknown rewards. In contrast, we consider linear MDP and non-myopic followers with bandit feedback and proposed a model-free RL algorithm. Recently, Kao et al. (2022) proposed a decentralized cooperative RL algorithm with a hierarchical information structure for a tabular set-up. In contrast, we consider a linear function approximation setup. Bai et al. (2021) provides a sample complexity guarantee for Stackelberg equilibrium for a bandit-RL game where the leader only takes action at the start of the episode which is quite different from our setup. Zheng et al. (2022) modeled the actor-critic framework as a Stackelberg game which is also quite different from our framework.

2. Leader-Follower MDP Game

The Model: We consider an episodic MDP with the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{B}, \mathcal{P}, H, \mathbf{R})$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space for leader, and $\mathcal{B}$ is the action space for followers. Each MDP starts from the state x_1. At every step h, observing the state x_h, the leader first takes an action $a_h \in \mathcal{A}$, then the follower takes an action $b_h \in \mathcal{B}$ observing the action of the leader and the state x_h. The state transitions to $x_{h+1} \in \mathcal{S}$ depending on x_h, a_h, and b_h. The process continues for H steps. The transition probability kernel is defined as the following, $\mathcal{P} = \{\mathbb{P}_h\}_{h=1}^{H}$ $\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathcal{S}$. The reward vector for leader and follower are defined as $\{r_{l,h}\}_{h=1}^{H}$ and $\{r_{f,h}\}_{h=1}^{H}$ respectively. The reward for agent $m = l$, and f, $r_{m,h}(x_h, a_h, b_h)$ denotes the reward received by agent m when the leader selects the action a_h and the follower selects the action b_h at step h.

Several important points on the model should be noted here. This setup is also known as hierarchical MDP (Kao et al., 2022) and can model various real-world applications. For example, consider a dynamic electricity market where a social planner sets a price at every hour of the day. Observing the price at the current period, the users decide how much to consume at that period. Here, the state can represent the demand of the user and the supply. The user's objective is to maximize its total utility over the day whereas the utility company's objective is to maximize the social welfare. Note that such kind of leader-follower decision hierarchy can also occur in MARL when one agent has advantage (e.g., can compute decisions faster compared to the other agent) over the other agent.

Throughout this paper, we consider a deterministic (unknown) reward. Without loss of generality, we also assume $|r_{m,h}| \leq 1$ for all m and h. Our model can be easily extended to the setup where the rewards are random yet bounded. Our model can also be extended to the setup where the initial state of each episodic MDP is drawn from a distribution.

Note that we consider the state x as a joint state of both the leader and the follower. Our work can be extended to the setup where the leader and the follower's states are decoupled with the underlying assumption that the leader can observe the follower's state.

Policy: The agents interact repeatedly over K episodes. The policy of the leader at step $h \in [H]$ at episode $k \in [K]$ is $\pi_{l,h}^{k}(a|x_h^k)$ that denotes the probability with which action $a \in \mathcal{A}$ is taken at step h at episode k when the state is x_h^k. The policy for the follower at step $h \in [H]$ at episode $[K]$ is $\pi_{f,h}^{k}(b|x_h^k, a_h^k)$ that denotes the probability with action $b \in \mathcal{B}$ is chosen by the follower at state x_h^k and when the leader's action is a_h^k. Note the difference with the simultaneous play, here, the follower's policy is a function of the leader's action at step h whereas in the simultaneous game, it is independent of the other players' actions. Let $\pi_l = \{\pi_{l,h}\}_{h=1}^{H}$, and $\pi_f = \{\pi_{f,h}\}_{h=1}^{H}$ be the collection of the policies of leader and follower respectively across the episode.

Q-function and Value function The joint state-action value function for any player m, for $m = l, f$ at step h is

$$Q_{m,h}^{\pi_l,\pi_f}(x,a,b) = \mathbb{E}\left[\sum_{i=h}^{H} r_{m,i}(x_i,a_i,b_i)|x_h = x, a_h = a, b_h = b\right]$$

Here, the expectation is taken over the transition probability kernel and the policies of both the leader and the follower. We can also define a *marginal* Q-function for the leader as

$$q_{l,h}^{\pi_l,\pi_f}(x,a) = \sum_{b} \pi_{f,h}(b|x,a) Q_{l,h}^{\pi_l,\pi_f}(x,a,b). \tag{1}$$

The above denotes the expected cumulative reward starting from step h after playing action a, and then following the policy π_l (from step $h+1$) while the follower following the policy π_f from step h. *We later show that marginal q plays an important role in decision-making.*

For the compactness of the operator, we introduce the following notations

$$\mathbb{D}^{\pi_l,\pi_f}[Q](x) = \mathbb{E}_{a\sim\pi_l(\cdot|x), b\sim\pi_f(\cdot|x,a)} Q(x,a,b), \qquad \mathbb{D}^{\pi_f}[Q](x,a) = \mathbb{E}_{b\sim\pi_f(\cdot|x,a)} Q(x,a,b),$$
$$\mathbb{P}_h V(x,a,b) = \mathbb{E}_{x'\sim\mathbb{P}_h(\cdot|x,a,b)} V(x').$$

The value function or expected cumulative reward starting from step h for the leader is defined as

$$V_{l,h}^{\pi_l,\pi_f}(x) = \mathbb{D}^{\pi_{l,h},\pi_{f,h}}[Q_{l,h}^{\pi_l,\pi_f}](x) \tag{2}$$

here the expectation is first taken over the follower's policy for a given leader's action then the expectation is taken over the leader's action.

We now consider the follower's value functions. The leader's action-dependent value function for the follower at step h when the leader takes action a at step h is given by

$$\bar{V}_{f,h}^{\pi_l,\pi_f}(x,a) = \mathbb{D}^{\pi_{f,h}}[Q_{f,h}^{\pi_l,\pi_f}](x,a) \tag{3}$$

The above denotes the expected cumulative reward the follower would get from step h after observing the leader's action at h, and then following its own policy at step h. We also define the following value function for the follower by taking the expectation over a on $\bar{V}$

$$V_{f,h}^{\pi_l,\pi_f}(x) = \mathbb{D}^{\pi_{l,h},\pi_{f,h}}[Q_{f,h}^{\pi_l,\pi_f}](x) \tag{4}$$

Bellman's Equation: We now describe the Bellman's equations for $m = l, f$ a given state and the joint action pair

$$Q_{m,h}^{\pi_l,\pi_f}(x,a,b) = r_{m,h}(x,a,b) + \mathbb{P}_h V_{m,h+1}^{\pi_l,\pi_f}(x,a,b) \tag{5}$$

Information Structure: We assume the following information structure–

Assumption 1 *The transition probability kernel, and rewards are unknown to both the leader and the follower. The leader and follower observe the rewards of each other only for the encountered state-action pairs (i.e,*bandit feedback*).*

Note that the agent can only access the rewards for the encountered states and actions. This is also known as the *bandit-feedback* setting (Bai et al., 2021). Thus, both the agents need to employ an exploratory policy. Also, note that we consider an information structure where the leader and follower can observe each other's reward. Extending our analysis to the setup where the leader does not observe the action and/or reward of the follower constitutes a future research direction.

Objective: After observing the action of the leader, the follower seeks to optimize its own leader's action-dependent value function: $\max_{\pi_f} \bar{V}_{f,1}^{\pi_l,\pi_f}(x_1, a)$

Given the policy of the follower, the leader seeks to optimize its own value function

$$\max_{\pi_l} V_{l,1}^{\pi_l,\pi_f}(x_1), \quad \pi_l^* = \arg\max_{\pi_l} V_{l,1}^{\pi_l,\pi_f}(x_1)$$

Now, we describe the relationship with Q-function and the value functions for the optimal policies of both the leader and the follower which will also specify how to select optimal policy at every step. For the follower, we have Bellman's optimality equation–

$$Q_{f,h}^{\pi_l,\pi_f^*}(x, a, b) = r_{f,h}(x, a, b) + \mathbb{P}_h V_{f,h+1}^{\pi_l,\pi_f^*}(x, a, b) \tag{6}$$

Hence, given the action a of the leader at step h, the follower's policy is $\pi_{f,h}^*(b|x, a) = 1$ where $b = \max_b Q_{f,h}^{\pi_l,\pi_f}(x, a, b)$. Thus, the optimal policy is greedy with respect to the joint state-action Q function. The follower's policy at step h depends on the leader's action at step h, the leader's policy and the follower's own policy starting from $h + 1$.

The optimal policy $\pi_{l,h}^*$ for the leader if the follower selects the policy π_f is given by

$$V_{l,h}^{\pi_l^*,\pi_f}(x_h) = \max_a q_{l,h}^{\pi_l^*,\pi_f}(x_h, a) \tag{7}$$

Hence, the optimal policy for the leader is greedy with respect to its marginal Q-function $q_{l,h}^{\pi_l^*,\pi_f}(x_h, \cdot)$. The optimal policy at step h depends on the follower's policy starting at step h, and the leader's policy starting from step $h + 1$.

If the Q-functions are known, the optimal policy for the leader and follower is obtained using backward induction. At step, H, the follower's best response for every leader's action is computed (as in (6)). Then, the leader's best response is computed based on the marginal q function (7). Once the policy is computed for step H, the policy for step $H - 1$ is computed in a similar manner and so on.

Learning Metric: Since the leader and the follower are unaware of the rewards and the transition probability, selecting an optimal policy from the start is difficult. Rather, they seek to learn policies with good performance guarantees. Thus, we consider the following learning metric:

Definition 1 *Regret for the leader is defined as*

$$\text{Regret}_\ell(K) = \sum_{k=1}^{K} (V_{l,1}^{\pi_l^{k,*},\pi_f^k}(x_1) - V_1^{\pi_l^k,\pi_f^k}(x_1)) \tag{8}$$

where $\pi_l^{k,*}$ is the optimal policy for the leader when the follower plays the policy π_f^k.

Note that the regret for the leader measures the optimality gap between the policies employed by the leader and the best response policy (in hindsight) of the leader given the follower's policy at an episode. Note that the follower's policy is unknown, rather, the leader needs to reason about the follower's policy. Also note that the leader's regret measures the gap between the policy π_l^k and the best policy $\pi_l^{k,*}$ of the leader in response to the follower's policy at episode k, rather than the optimal policy of the follower. This is because the follower is also learning its optimal policy, hence, the policy π_f^k may not be optimal. Thus, the regret considers how good the leader is doing compared to the policy employed by the follower across the episodes. Obviously, at different episodes, the optimal policy of the leader may be different.

Definition 2 *Regret for the follower is defined as*

$$\text{Regret}_f(K) = \sum_{k=1}^{K}(\bar{V}_{f,1}^{\pi_l,\pi_f^*}(x_1, a_1^k) - \bar{V}_{f,1}^{\pi_l,\pi_f^k}(x_1, a_1^k)) \tag{9}$$

where π_f^* denotes the optimal policy for the follower and a_1^k is the action of the leader at the first step of episode k.

The regret for the follower captures how good the follower's policy is compared to the optimal policy for a given initial action and the policy of the leader. Even though the follower knows the initial action of the leader, it is unaware of the leader's policy. Rather, the follower also needs to reason about the leader's policy. Both the leader and the follower seek to minimize their respective regrets. *Initial action-specific regret is unique to the leader-follower setup.*

Note that Zhong et al. (2021) considers the setup where the leader takes action at every step of the MDP whereas the followers are myopic. They also consider that the rewards are known, thus, one can compute the best response of the follower at every step. Thus, the regret for the follower does not arise there. Kao et al. (2022) considered the reward is the same, thus, they consider the regret of the joint policy rather than the individual agent's regret.

In general, achieving sub-linear regret in a multi-agent RL setup is more challenging compared to the single-agent setup since the underlying environment of an agent may change depending on the other agent's policy (Tian et al., 2021; Bai et al., 2021). Nevertheless, we obtain sub-linear regret for both the leader and follower.

Leader-follower Linear MDP: We consider a linear MDP set-up in order to handle large state-space.

Assumption 2 *We consider a linear MDP with the known (to both the players) feature map $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}^d$, if for any h, there exists d* unknown *signed measures $\mu_h = \{\mu_h^1, \ldots, \mu_h^d\}$ over $\mathcal{S}$ such that for any $(x, a, x') \in \mathcal{S} \times \mathcal{A}_1 \times \mathcal{B}$,*

$$\mathbb{P}_h(x'|x, a, b) = \langle \phi(x, a, b), \mu_h(x') \rangle$$

and there exists vectors $\theta_{l,h}$, $\theta_{f,h} \in \mathbb{R}^d$ such that for any $(x, a, b) \in \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2$,

$$r_{l,h}(x, a, b) = \langle \phi(x, a, b), \theta_{l,h} \rangle \quad r_{f,h}(x, a) = \langle \phi(x, a, b), \theta_{f,h} \rangle$$

Note that such a linear MDP setup is considered for single agent scenario Jin et al. (2020); Yang and Wang (2019). Examples of linear MDP include tabular setup. Our analysis can be easily extended to the setup where the followers and leaders have different feature spaces (i.e., $r_{m,h} = \langle \phi_m, \theta_{m,h} \rangle$, for transition probability, we can concatenate the feature-space $\phi = [\phi_l, \phi_f]^T$).

For the leader-follower linear MDP setup, we have–

Lemma 1 $Q_{m,h}^{\pi_l,\pi_f}(x, a, b) = \langle \phi(x, a, b), w_{m,h}^{\pi_l,\pi_f} \rangle \; \forall m, x, a, b, h.$

Thus, the Q-functions of both the leader and follower are linear in the feature space. We can thus search over w in order to find the optimal Q-function.

Algorithm 1 Leader's Model Free RL Algorithm

```
1: Initialization: $w_{l,h} = 0,\ w_{f,h} = 0,\ \alpha_f = \log(|\mathcal{B}|)\sqrt{K}/H,\ \alpha_l = \log(|\mathcal{A}|)\sqrt{K}/H, \beta = C_1 dH\sqrt{\log(4(\log(|\mathcal{B}||\mathcal{A}|) + 2\log(|\mathcal{B}|)\log(|\mathcal{A}|))dT/p)}$
2: for episodes $k = 1, \dots, K$ do
3:     Receive the initial state $x_1^k$.
4:     for step $h = H, H-1, \dots, 1$ do
5:         $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)\phi(x_h^\tau, a_h^\tau, b_h^\tau)^T + \lambda\mathbf{I}$
6:         $w_{m,h}^k \leftarrow (\Lambda_h^k)^{-1}[\sum_{\tau=1}^{k-1} \phi(x_h^\tau, a_h^\tau, b_h^\tau)[r_{m,h}(x_h^\tau, a^\tau, b_h^\tau) + V_{m,h+1}^k(x_{h+1}^\tau)]]$
7:         $Q_{m,h}^k(\cdot,\cdot,\cdot) \leftarrow \min\{\langle w_{m,h}^k, \phi(\cdot,\cdot,\cdot)\rangle + \beta(\phi(\cdot,\cdot,\cdot)^T(\Lambda_h^k)^{-1}\phi(\cdot,\cdot,\cdot))^{1/2}, H\}$
8:         for $a \in \mathcal{A}$ do
9:             $\pi_{f,h}^k(b|\cdot, a) = \frac{\exp(\alpha_f(Q_{f,h}^k(\cdot, a, b)))}{\sum_{b_m} \exp(\alpha_f(Q_{f,h}^k(\cdot, a, b_m)))}$
10:            $q_{l,h}^k(\cdot,\cdot) \leftarrow \langle \pi_{f,h}^k(\cdot|\cdot,\cdot), Q_{l,h}^k(\cdot,\cdot,\cdot)\rangle$
11:            $\bar{V}_{f,h}^k(\cdot,\cdot) \leftarrow \langle \pi_{f,h}^k(\cdot|\cdot,\cdot), Q_{f,h}^k(\cdot,\cdot,\cdot)\rangle$
12:        $\pi_{l,h}^k(a|\cdot) = \frac{\exp(\alpha_l(q_{l,h}^k(\cdot, a)))}{\sum_{a_m} \exp(\alpha_l(q_{l,h}^k(\cdot, a_m)))}$
13:        $V_{l,h}^k(\cdot) = \langle \pi_{l,h}^k, q_{l,h}^k(\cdot,\cdot)\rangle$, $V_{f,h}^k(\cdot) = \langle \pi_{l,h}^k, \bar{V}_{f,h}^k(\cdot,\cdot)\rangle$
14:    for $h = 1, \dots, H$ do
15:        for $a \in \mathcal{A}$ do
16:            Compute $Q_{f,h}^k(x_h^k, a, b)$, $Q_{l,h}^k(x_h^k, a, b)$ for all $b$.
17:            Compute policy $\pi_{f,h}^k(b|x_h^k, a)$ according to the Soft-max for $Q_{f,h}^k(x_h^k, a, \cdot)$ with parameter $\alpha_f$.
18:            $q_{l,h}(x_1^k, a) = \sum_b \pi_{f,h}^k(b|x_h^k, a)Q_{l,h}^k(x_h^k, a, b)$
19:        Leader takes action $a_h^k$ according to Soft-max policy with respect to $q_{l,h}^k(x_h^k, \cdot)$ with parameter $\alpha_l$.
20:        The follower takes an action $b_h^k$ (Algorithm 2) and observe $x_{h+1}^k \sim \mathbb{P}_h(x_h^k, a_h^k, b_h^k)$
```

Algorithm 2 Follower's Model Free RL Algorithm

```
1: Execute steps 1-14 of Algorithm 1.
2: for step $h = 1, \dots, H$ do
3:     Observe the action $a_h^k$ of the leader.
4:     Compute $Q_{f,h}^k(x_h^k, a_h^k, b)$, $\pi_{f,h}^k(b|x_h^k, a^k)$ for all $b$ based on $w_{f,h}^k$.
5:     The follower takes action $b_h^k \sim \pi_{f,h}^k(\cdot|x_h^k, a_h^k)$ and observe $x_{h+1}^k$.
```

3. Proposed Algorithm

We now describe our proposed algorithms for the leader (Algorithm 1) and the follower (Algorithm 2). The algorithm is based on the LSVI-UCB Jin et al. (2020) with some subtle differences which we will point out in our description.

We first describe the leader's algorithm. Note that in order to obtain its policy, the leader also needs to reason the policy the follower would play. Hence, the leader's algorithm also consists of how the follower selects its policy. The first part (steps 5-6) consists of updating the parameters $\Lambda_h^k, w_{l,h}^k, w_{f,h}^k$ which are used to update the joint state-action value functions $Q_{m,h}^k$ and value functions $V_{m,h}^k$ for $m = l, f$. Note that Steps 7-14 are not evaluated for each state, rather, they are evaluated only for the encountered states till episode $k-1$. Hence, we do not need to iterate over a potentially infinite number of states. $V_{m,H+1}^k = 0$ for all k.

We now discuss the rationale behind updating $w_{m,h}^k$. We seek to obtain w such that it approximates the Q-function since the Q-function is the inner product of w and ϕ (Lemma 1). Thus, we parameterize $Q_{m,h}^{\pi_l,\pi_f}(\cdot,\cdot,\cdot)$ by a linear form $\langle w_{m,h}^k, \phi(\cdot,\cdot,\cdot)\rangle$. The intuition is to obtain $w_{m,h}^k$ from Bellman's equation using the regularized least-square regression. However, there are challenges.

We do not know $\mathbb{P}_h$ in Bellman's equation (5) rather $\mathbb{P}_h V_{m,h+1}^{\pi_l;\pi_f}$ should be replaced by the empirical samples. We obtain $w_{m,h}^k$ for $m = l, f$ by solving the following regularized least-square problem

$$w_{m,h}^k \leftarrow \arg\min_{w \in \mathbb{R}^d} \sum_{\tau=1}^{k-1} [r_{m,h}(x_h^\tau, a_h^\tau, b_h^\tau) + V_{m,h+1}^k(x_{h+1}^\tau) - w^T \phi(x_h^\tau, a_h^\tau)]^2 + \lambda ||w||_2^2 \tag{10}$$

where $V_{m,h+1}^k$ is the estimate of the value function $V_{m,h+1}$. After we obtain $w_{m,h}^k$, we add an additional bonus term $\beta(\phi(\cdot,\cdot,\cdot)^T(\Lambda_h^k)^{-1}\phi(\cdot,\cdot,\cdot))^{1/2}$ similar to Jin et al. (2020) to obtain $Q_{m,h}^k$. β is constant which we will characterize in the next section. Λ_h^k is the Gram matrix for the regularized least square problem. Such an additional term is used for the upper confidence bound in LSVI-UCB Jin et al. (2020) as well. The same additional term is used for both $Q_{l,h}^k$ and $Q_{f,h}^k$. This bonus term would ensure the exploration for both the leader and the follower.

Now, we describe how we estimate the value function $V_{m,h+1}^k$ function which we use in (10). In order to update the value function, we need to compute the policy for the follower and the leader (cf.(2)). Unlike LSVI-UCB, we use the soft-max policy for both the leader and follower. Soft-max policy $\text{SOFT-MAX}_\alpha(\mathbf{X}) = \{\text{SOFT-MAX}_\alpha^i(\mathbf{X})\}_{i=1}^{|\mathcal{L}|}$ for any vector $\mathbf{X} \in \mathbb{R}^{|\mathcal{L}|}$ is a vector with the same dimension as in $\mathbf{X}$ with parameter α where the i-th component

$$\text{SOFT-MAX}_\alpha^i(\mathbf{X}) = \frac{\exp(\alpha X_i)}{\sum_{n=1}^{|\mathcal{L}|} \exp(\alpha X_n)} \tag{11}$$

In order to estimate the value function, first, one needs to compute follower's policy at a given leader's action, and subsequently, the leader's policy needs to be computed (cf.(2) & (4)). At step h, for every leader's action a, $\pi_{f,h,k}(b|x_h^\tau, a)$ is computed based on the soft-max policy on the estimated Q-function for the follower $Q_{f,h}^k(x_h^\tau, a, b)$ at step 9. Based on the follower's policy, the leader updates its marginal Q function q_l at step 10 (cf.(1)) and the follower's leader's action dependent value function $\bar{V}_{f,h}^k$ (cf.(3)) at step 11. Now, the leader computes its policy based on $q_{l,h}^k(\cdot,\cdot)$. Finally, We update the leader's and follower's value function based on the leader's policy at steps 13 and 14.

Why Soft-max?: Note that when $\alpha_l = \alpha_f = \infty$, the policy of the follower and leader become equal to the greedy policy. The greedy policy is optimal for the leader with respect to its marginal q-function (Eq.(7)) and for the follower with respect to its joint Q-function (cf.(6)). However, we can not obtain sub-linear regret for the leader and follower with the greedy policy unlike the single agent scenario. In particular, the greedy policy is not Lipschitz, hence, it does not provide uniform concentration bound for each agent's value function, an essential step in the regret bound (Section 4).

The last part consists of the execution of the policy. In order to find its optimal policy, the leader computes $Q_{f,h}^k$ for each action of the leader a (Line 17), based on the already computed $w_{f,h}^k$. Once $Q_{f,h}^k$ is computed, the follower's policy is also computed based on the soft-max function (Step 18). Once the follower's policy is computed, the marginal Q-function for the leader $q_{l,h}^k(x_h^k,\cdot)$ is computed. The leader then takes an action a_h^k based on the soft-max policy with respect to $q_{l,1}^k$. The follower takes an action b_h^k by observing the action a_h^k which we will describe next.

As mentioned before, the steps of the follower (Algorithm 2) are already contained in the leader's algorithm. The follower also obtains its w by solving (10). Hence, the leader and the follower have the same updates on w. The only difference is the execution as the follower executes its action based on the action taken by the leader at every step. At step 4 of Algorithm 2, the follower computes Q-function based on the current state x_h^k and action a_h^k of the leader. The follower then chooses its action based on the soft-max policy on the Q function.

Space and Time Complexity: The space and time complexities of Algorithms 1 and 2 are of the same order as the LSVI-UCB. To be precise, the space complexity is $\mathcal{O}(d^2H + d|\mathcal{A}||\mathcal{B}|T)$. When we compute $(\Lambda_h^k)^{-1}$ using Sherman-Morrison formula, the computation of $V_{m,h+1}^k$ is dominated by computing $Q_{m,h+1}^k$ and the policy π_l^k, and π_f^k. Hence, it takes $\mathcal{O}(d^2|\mathcal{A}||\mathcal{B}|T)$ time.

4. Main Results

Theorem 3 *Fix $p > 0$. If we set $\beta = C_1 dH\sqrt{\iota}$ in Algorithm 1 and Algorithm 2 where $\iota = \log((\log(|\mathcal{A}||\mathcal{B}|) + 2\log(|\mathcal{A}|)\log(|\mathcal{B}|))4dT/p)$ for some absolute constant C_1, then w.p. $(1-p)$,*

$$\text{Regret}_l(K) \leq C\sqrt{d^3H^3T\iota^2}, \qquad \text{Regret}_f(K) \leq C'\sqrt{d^3H^3T\iota^2}$$

where $T = KH$ for some absolute constants C, and C'.

The result shows that regret for both the leader and the follower scale with $\tilde{\mathcal{O}}(\sqrt{d^3H^3T})$. Note that for the single-agent scenario (Jin et al., 2020), the order of the regret is the same. However, compared to Jin et al. (2020), there is an additional multiplicative $\log(|\mathcal{A}|)$ and $\log(|\mathcal{B}|)$ factor in the value of ι which arises because we use soft-max policy for the leader and the follower instead of the greedy policy. The regret bounds do not depend on the dimension of the state space, rather, it depends on the dimension of the feature space d. If there is no follower, we set $|\mathcal{B}|= 1$, and can achieve a single agent's regret for the soft-max policy as well. *To the best of our knowledge this is the first result which shows $\tilde{\mathcal{O}}(\sqrt{T})$ regret for both the leader and the follower in the model-free with function approximation.*

We assume that the leader observes the follower's action (this is known as informed game (Tian et al., 2021)). Under an uninformed setting (where a player may not observe the action of other), it is statistically hard to obtain sub-linear regret even in zero-sum game (Tian et al., 2021). Thus, Tian et al. (2021) shows a sub-linear regret under a weaker notion of regret. It remains to be seen under such information structure, whether such a result holds in the leader-follower setup.

Please see Section 4.2 of the technical report Ghosh (2022) for the outline of the proof and the Appendix for the detailed proof. In the following, we provide the intuition on why we need a soft-max policy.

Why Soft-max: A key step in proving the regret is to show the following

$$\left\|\sum_{\tau=1}^{k-1}\phi(x_h^\tau, a_h^\tau, b_h^\tau)\left[V_{m,h+1}^k(x_{h+1}^\tau) - \mathbb{P}_h V_{m,h+1}^k(x_h^\tau, a_h^\tau, b_h^\tau)\right]\right\|_{(\Lambda_h^k)^{-1}}$$

is upper bounded by $O(d\log K)$ for both $m = l, f$. In order to prove the above, one uses the uniform concentration bound as done in Jin et al. (2020) for single-agent case.

In particular, we need to show that log ϵ-covering number for value function class $(V_{m,h+1}^k)$ for both the leader and the follower scales at most $\log(K)$. In single agent, ϵ-covering number for Q-function was enough as max is a contraction operator. However, in our setting, the value function for each agent inherently depends on the policy of the other agent in our setup. Thus, if the policy is greedy, slight change in the Q-function for one agent could lead to a substantial change in the value function for the other agent and vice versa (please see the example in Appendix F in Ghosh (2022)). Thus, one can not obtain log ϵ-covering number for the value function which scales at most $\log(K)$ for greedy policy even though the log ϵ-covering number for Q function scales at most $\log(K)$. Using the Lipschitz property of the Soft-max we show that indeed the log ϵ-covering number for both the leader and the follower scale at most $\log(K)$ (Lemma 15 in our technical report Ghosh (2022)). In particular, the soft-max policy of parameter α results into a Lipschitz constant of 2α for the policy which would ensure that individual value functions are also also Lipschitz in the Q

functions which enables us to obtain the desired bound on the covering number of value function class. Please see Section 4.2 in Ghosh (2022) for detail.

Of course, one would ask the question why not make the temp. co-efficient (α) smaller which would result in a smaller Lipschitz constant. However, the performance would be poor with respect to the greedy policy. Scaling the temp. coefficient with $O(K)$ achieves the trade-off. Recently, Ghosh et al. (2022) also uses soft-max to obtain sub-linear regret in different setup as well (constrained linear MDP, i.e., min-max setup).

Remark 4 *We have assumed that the feature space ϕ is known. Note that feature space learning is an active area of research (Agarwal et al., 2020; Modi et al., 2021) for linear approximation setup. The most promising technique is to estimate the Q-function by jointly optimizing over w and ϕ. Neural networks can be used to obtain such w and ϕ. Using similar technique we can learn ϕ, however, such a characterization is left for the future.*

5. Equilibrium Learning

The Algorithms 1 and 2 also enables us to obtain equilibrium policies (ϵ-close).

Finding Coarse Correlated Stackelberg Equilibrium (CCSE , Definition 6 in Ghosh (2022)):

Corollary 1 *Consider the joint policy: the leader and follower jointly choose a $k \in [K]$ with prob. $1/K$, and then the leader and follower select the policy π_l^k and π_f^k respectively (returned by Algorithms 1 and 2). Such a joint policy is $\tilde{\mathcal{O}}(\sqrt{d^3H^4/K})$-CCSE with probability $1-p$.*

The above result entails that in order to achieve ϵ-CCSE one needs $\tilde{\mathcal{O}}(1/\epsilon^2)$ episodes. This is the first such result for a non-myopic leader and follower Markov game setup with function approximation. The agent only needs to coordinate on the random number to choose the episode index k. The proof is in Appendix H of the technical report Ghosh (2022).

Finding Stackelberg Equilibrium: For a zero-sum game, CCSE coincides with the Stackelberg equilibrium. Hence, by Corollary 1, we also obtain ϵ- Stackelberg equilibrium for a zero-sum game.

For a more general setting, we can combine the reward-free exploration proposed in Wang et al. (2020); Liu et al. (2021) and the soft-max policy to obtain SE with self-play. We divide the total episodes into two phases: exploration, and exploitation. During the exploration phase, both the leader and the follower explore to reduce the confidence bound. Instead of a true reward, the reward will be the bonus term, $||\phi(x,a,b)||_{(\Lambda_h^k)^{-1}}$ which will incentivize the players to explore similar to Wang et al. (2020) (using soft-max policy instead of greedy policy). In the exploitation phase, the leader and follower adopt policies similar to Algorithms 1 and 2.

6. Conclusion and Future Work

We propose a model-free RL-based algorithm for both the leader and the follower. We have achieved $\tilde{\mathcal{O}}(\sqrt{d^3H^3T})$ regret for both the leader and the follower. We have extended the LSVI-UCB algorithm towards the leader-follower setup. We have underlined the technical challenges in doing so and explained how the greedy policy for the players fails to achieve a uniform concentration bound for the individual value function. We show that a soft-max policy for the players can achieve the regret bound. We also obtain the convergence rate of CCSE equilibrium.

Whether we can tighten this dependence on d and H remain an important future research direction. Finally, we consider one leader and one follower setup. Extending our setup to multiple leaders and followers constitutes an important future research direction. Extending our setup to non-linear function approximation also constitutes a future research direction. In this regard, we leverage on the recent works on simultaneous games with general function approximation Jin et al. (2022); Wang et al. (2023); Li et al. (2022).

Acknowledgment

Arnob Ghosh acknowledges Ness Shroff and Xingyu Zhou for an initial discussion of this work. This work of Arnob Ghosh has been partly supported by NSF grants NSF AI Institute (AI-EDGE) 2112471.

References

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*, pages 551–560. PMLR, 2020.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

Yu Bai, Chi Jin, Huan Wang, and Caiming Xiong. Sample-efficient learning of stackelberg equilibria in general-sum games. *Advances in Neural Information Processing Systems*, 34:25799–25811, 2021.

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015.

Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014.

Zixiang Chen, Dongruo Zhou, and Quanquan Gu. Almost optimal algorithms for two-player markov games with linear function approximation. *arXiv preprint arXiv:2102.07404*, 2021.

Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 243–252, 2014.

Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 132–141, 2004.

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

Arnob Ghosh. Provably efficient model-free rl in leader-follower mdp with linear function approximation, 2022. URL https://arxiv.org/abs/2211.15792.

Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Provably efficient model-free constrained rl with linear function approximation. In *36th Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021.

Chi Jin, Qinghua Liu, and Tiancheng Yu. The power of exploiter: Provable multi-agent rl in large state spaces. In *International Conference on Machine Learning*, pages 10251–10279. PMLR, 2022.

Hsu Kao, Chen-Yu Wei, and Vijay Subramanian. Decentralized cooperative reinforcement learning with hierarchical information structure. In *International Conference on Algorithmic Learning Theory*, pages 573–605. PMLR, 2022.

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *International symposium on algorithmic game theory*, pages 250–262. Springer, 2009.

Chris Junchi Li, Dongruo Zhou, Quanquan Gu, and Michael Jordan. Learning two-player markov games: Neural function approximation and correlated equilibrium. *Advances in Neural Information Processing Systems*, 35:33262–33274, 2022.

Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021.

Weichao Mao and Tamer Başar. Provably efficient reinforcement learning in decentralized general-sum markov games. *Dynamic Games and Applications*, pages 1–22, 2022.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2149–2156, 2019.

Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized q-learning in zero-sum markov games. *Advances in Neural Information Processing Systems*, 34:18320–18334, 2021.

Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press, 2011.

Yi Tian, Yuanhao Wang, Tiancheng Yu, and Suvrit Sra. Online learning in unknown markov games. In *International conference on machine learning*, pages 10279–10288. PMLR, 2021.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

Yuanhao Wang, Qinghua Liu, Yu Bai, and Chi Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent rl with function approximation. *arXiv preprint arXiv:2302.06606*, 2023.

Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Online reinforcement learning in stochastic games. *Advances in Neural Information Processing Systems*, 30, 2017.

Chen-Yu Wei, Mehdi Jafarnia Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, pages 10170–10180. PMLR, 2020.

Qiaomin Xie, Yudong Chen, Zhaoran Wang, and Zhuoran Yang. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. *arXiv preprint arXiv:2002.07066*, 2020.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Liyuan Zheng, Tanner Fiez, Zane Alumbaugh, Benjamin Chasnov, and Lillian J Ratliff. Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9217–9224, 2022.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.

Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopic followers? *arXiv preprint arXiv:2112.13521*, 2021.