Policy Learning for Robust Markov Decision Process with a Mismatched Generative Model

Jialian Li[1], **Tongzheng Ren**[2], **Dong Yan**[1], **Hang Su**[1], **Jun Zhu**[1*]

[1] Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Tsinghua-Bosch Joint Center for Machine Learning, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China

[2] Department of Computer Science, UT Austin

lijialian7@163.com, tongzheng@utexas.edu, sproblvem@gmail.com, suhangss@tsinghua.edu.cn, dcszj@tsinghua.edu.cn

Abstract

In high-stake scenarios like medical treatment and auto-piloting, it's risky or even infeasible to collect online experimental data to train the agent. Simulation-based training can alleviate this issue, but may suffer from its inherent mismatches from the simulator and real environment. It is therefore imperative to utilize the simulator to learn a robust policy for the real-world deployment. In this work, we consider policy learning for Robust Markov Decision Processes (RMDP), where the agent tries to seek a robust policy with respect to unexpected perturbations on the environments. Specifically, we focus on the setting where the training environment can be characterized as a generative model and a constrained perturbation can be added to the model during testing. Our goal is to identify a near-optimal robust policy for the perturbed testing environment, which introduces additional technical difficulties as we need to simultaneously estimate the training environment uncertainty from samples and find the worst-case perturbation for testing. To solve this issue, we propose a generic method which formalizes the perturbation as an opponent to obtain a two-player zero-sum game, and further show that the Nash Equilibrium corresponds to the robust policy. We prove that, with a polynomial number of samples from the generative model, our algorithm can find a near-optimal robust policy with a high probability. Our method is able to deal with general perturbations under some mild assumptions and can also be extended to more complex problems like robust partial observable Markov decision process, thanks to the game-theoretical formulation.

Introduction

Reinforcement Learning (RL) (Sutton, Barto et al. 1998) aims to identify good policies that can solve the sequential decision-making problem, and has achieved amounts of incredible results on different challenging tasks (Mnih et al. 2013, 2015; Silver et al. 2017). Most of the current work focuses on the case where we can evaluate the performance of learned policies on the training environments, which is reasonable in multiple cases. For example, we can evaluate the policy for video games under the same environment since the environmental dynamics or game rules are the same for both training and testing.

However, most of these algorithms need to interact with the environment and collect online experimental data to train the agent, which restricts the application of these algorithms in high-stake scenarios like medical treatment and auto-piloting. In certain cases, we have access to an additional simulator (e.g. for auto-piloting we can train the agent in a simulator that simulates the real traffic scenarios) which allows us to train the agent on the simulator. However, the simulator can substantially suffer from the domain mismatch from the real application scenarios, which may lead to a significant performance degeneration when deploying the model in real world.

We can instead utilize this imperfect simulator for training and find a relatively robust policy such that it can also work well in the real world. This indeed falls into the distributional robust optimization (DRO) problem (Rahimian and Mehrotra 2019).

In this work, we consider DRO problems where the environment can be formalized as a Markov Decision Process (MDP) with finite steps. Several existing works (Nilim and El Ghaoui 2004; Iyengar 2005) treat the difference of the training and testing MDP as a perturbation and the goal is to find a robust policy that ensures a near-optimal reward against the worst-case perturbation, which is so-called a Robust MDP (RMDP) problem. Many previous works (e.g. Nilim and El Ghaoui 2004; Iyengar 2005) assume that all the parameters of the training MDP and the perturbation set are known. Then the robust policy can be solved directly by (approximated) dynamic programming. However, in the practical problems, the simulator may be constructed as a complex system with physical computation (Comair et al. 2014; Rasheed, San, and Kvamsdal 2020) which can only provide samples through interactions. Other works (Delage and Mannor 2010; Burges et al. 2013; Ghavamzadeh, Petrik, and Chow 2016) attempt to address problems under parameter uncertainty, but they generally neither take extra perturbation into consideration, nor provide sample complexity analysis for a near-optimal result.

Solving RMDPs in an online manner is quite hard, because we might not be able to reach some potentially important states during online training. This might lead to policy's bad performance during testing. Jiang (2018) gives an extreme case where there is only one single state-action pair difference between the simulator and real environment,

*corresponding author.

which makes the learned policy information-theoretically useless. To the best of our knowledge, there are no satisfactory results for solving RMDP in an online fashion. Jiang (2018) assumes that it has access to the testing environment and can get data to correct training simulator. However, we want to remark that such cases may not be applicable in numerous practical scenarios, as we cannot take such risk to collect online data in the testing environment. Roy, Xu, and Pokutta (2017) use model-free methods to solve the RMDP problem by assuming the perturbation set fixed and given. However, they do not provide the non-asymptotic sample complexity. In some bad cases of online RMDP, where some potential important states are hard to reach, the algorithm may suffer a large sample complexity.

In this work, we consider the online training RMDP problems in a relaxed way. We assume the training environment is characterized via a generative model and the agent queries the generative model to gain samples from the training MDP. Although solving an MDP with a generative model has gained much interest (Kearns and Singh 1999; Azar, Munos, and Kappen 2013; Cui and Yang 2020), the corresponding analysis in RMDP remains mostly open problem. We extend the fix perturbation set assumption in (Roy, Xu, and Pokutta 2017) to general constraints that the perturbation may be dependent on the environment parameters. This would lead to a new challenge that our estimation error for the training environment can further influence the robust policy solving.

Our Contributions. To address the aforementioned issues, we propose a method for solving RMDPs under general constraints. We first re-formulate the RMDP problem as a two-player zero-sum game by considering the perturbation as an adversarial player, and show that the Nash Equilibrium (NE) of the game corresponds to the robust policy and worst-case perturbation. Then we propose a new algorithm that uses a plug-in NE solver to find the robust policy.

We provide rigorous non-asymptotic sample complexity bounds for our algorithm in certain cases. For RMDP with S states, A actions, and horizon H, we give non-asymptotic analysis for constraints with the Lipschitz condition. We show that with samples of order $\widetilde{O}((1+\lambda)^2 S^2 A H^4/\epsilon^2)$[1], our learned policy has an error no larger than ϵ with a high probability, where λ is the Lipschitz constant.

We finally remark that, since we use a game-theoretical framework to consider the environment uncertainty and robust policy solving separately, we can extend our method to problems with other uncertainty. We give an example on solving the robust partial-observable MDP problems.

Problem Formulation

In this section, we introduce the background of Markov Decision Process (MDP) and robust Markov Decision Process (RMDP) problem in this work.

Markov Decision Process (MDP)

We consider the finite-horizon Markov Decision Process (MDP) $M = \langle \mathcal{S}, \mathcal{A}, P, r, H \rangle$ where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. We denote $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$. Transition probability $P(s,a)$ represents the probability distribution of transiting to states in $\mathcal{S}$ if taking action a at state s. Reward function r maps a state-action pair (s,a) to a reward value $r(s,a) \in [0,1]$. One episode of interaction has a depth of H which is denoted as $[H] = \{1, 2, ..., H\}$.

Usually for finite-horizon MDPs, the optimal actions for one state at different depths can be different. For clarity, we assume the state sets for different depths are disjoint. Formally, we use $\mathcal{S}_h$ to denote the set of states with depth h and $D = \max_h |\mathcal{S}_h|$, where $\mathcal{S} = \cup_{h=1}^{H} \mathcal{S}_h$ and $\mathcal{S}_h \cap \mathcal{S}_{h'} = \emptyset$ if $h \neq h'$. Furthermore we denote the simplex of all possible transition probability vectors on $P(s,a)$ as Δ_{sa}.

For clarity, we assume an initial state s_0 at depth 1. At state $s \in \mathcal{S}_h$, the agent chooses one action $a \in \mathcal{A}$. Then the environment turns to state $s' \in \mathcal{S}_{h+1}$ with a probability of $P(s'|s,a)$. After H times of interactions from depth 1, one episode ends. For the convenience of notation, we define a terminal state $s_{\mathcal{T}}$ at depth $H+1$ and the environment evolves to $s_{\mathcal{T}}$ from depth H.

A policy π of the agent maps each state $s \in \mathcal{S}$ to an action in $\mathcal{A}$. Usually, each state's policy can be a distribution over $\mathcal{A}$. For a finite horizon MDP, there always exists a deterministic optimal policy(Puterman 2014). Here without further clarification, our "policy" indicates a deterministic policy for the agent.

Consider state $s \in \mathcal{S}_h$, we use $V_M^\pi(s)$ to represent the expected reward, following the policy π in MDP M with transition P:

$$V_M^\pi(s) = \mathbb{E}_P \left[\sum_{h'=h}^{H} r(s_{h'}, \pi(s_{h'})) | s_h = s \right],$$

where $\mathbb{E}_P$ indicates the expectation over transition function. Similarly, we define the Q values for state-action pairs as

$$Q_M^\pi(s,a) = \mathbb{E}_P \left[r(s,a) + \sum_{h'=h+1}^{H} r(s_{h'}, \pi(s_{h'})) \right].$$

The Bellman equation for MDP $M = \langle \mathcal{S}, \mathcal{A}, P, r, H \rangle$ at state $s \in \mathcal{S}_h$ is

$$V_M^\pi(s) = r(s, \pi(s)) + \sum_{s' \in \mathcal{S}_{h+1}} P(s'|s,a) V_M^\pi(s').$$

Specifically, we define $V_M^\pi(s_{\mathcal{T}}) = 0$ for any π or M. Our final goal is to find a policy π' such that

$$\pi' = \operatorname*{argmax}_\pi V_M^\pi(s_0).$$

Robust Markov Decision Process (RMDP)

We consider the problem where the environments at training and testing phases can be different. Denote the testing MDP as $M^* = \langle \mathcal{S}, \mathcal{A}, P^*, r, H \rangle$ and the training MDP as $M^s = \langle \mathcal{S}, \mathcal{A}, P^s, r, H \rangle$. For clarity, here we only assume the transitions are different between M^* and M^s, while our techniques to handle the transition can be easily generalized to handle the rewards.

During the training time, P^* is not available for the agent and only samples from P^s are accessible via a generative

[1] The notation $\widetilde{O}$ represents the order ignoring logarithm term.

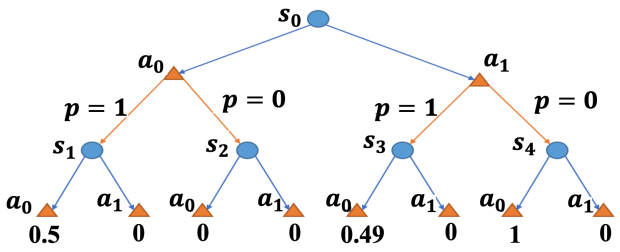

Figure 1: The structure of the MDP

model. At each time step, the agent sends a state-action pair (s, a) to the generative model and gains $(s', r(s, a))$, where s' is sampled from $P^s(s, a)$. To enable a policy to transfer from the training environment to the testing environment, P^s and P^* should be to some degree similar. Here we use a constraint to describe their connection. We use $\prod$ to denote the Cartesian product and define $\mathcal{P} = \prod_{(s,a)} \Delta_{sa}$ to be the set composed of all the transition probabilities. Here denote dimension of $\mathcal{P}$ to be $G \leq SAD$. Then we assume a known constraint function $\mathcal{C}$ which maps each $P \in \mathcal{P}$ to a set $\mathcal{C}(P) \subseteq \mathcal{P}$, and also assume that the testing transition function P^* locates in $\mathcal{C}(P^s)$. Further we subtract the transition P from $\mathcal{C}(P)$ to define the perturbation set as

$$U(P) := \{p \in [-1, 1]^G : P + p \in \mathcal{C}(P)\}. \quad (1)$$

If $U(P^s)$ contains only elements near 0 (i.e., $\|p\|$ is small), then P^* is constrained to be within a neighborhood of P^s. Furthermore, we use $\mathcal{C}(M)$ to denote the set of MDPs whose transition functions are located in $\mathcal{C}(P)$.

In order to learn a robust policy, we need to consider the worst-case environment for each policy. We define a worst-case V value for policy π as

$$\widetilde{V}^{\pi}(s) = \min_{M \in \mathcal{C}(M^s)} V_M^{\pi}(s), \quad (2)$$

and the optimal robust policy π^* can be defined as

$$\pi^* = \operatorname*{argmax}_{\pi} \widetilde{V}^{\pi}(s_0).$$

For a policy π, we define the error between π and π^* as

$$Err(\pi) := \widetilde{V}^{\pi^*}(s_0) - \widetilde{V}^{\pi}(s_0). \quad (3)$$

Our learning goal is to find a policy π' such that

$$\mathbb{P}(Err(\pi') \leq \epsilon) \geq 1 - \delta, \quad (4)$$

for some $\delta \in (0, 1)$ and $\epsilon > 0$. Here $\mathbb{P}(\cdot)$ denotes the probability for some event. A key for this learning target is that the optimal policy for M^s might work poorly under M^*, which urges for a robust policy for testing phase.

A Simple Case for Intuition

Here we give an simple example to show that robustness is indeed an import issue under the model mismatch problems.

We consider a training MDP M^s with only 2 layers. Depth 1 has only one state, i.e. s_0 and depth 2 has four states. The agent has two actions a_0 and a_1 at each state. The agent receives a reward at the end of one episode. The structure of the MDP and the rewards are given in Fig. 1. It easy to see that the optimal strategy for M^s is always to choose a_0 and the optimal reward is 0.5.

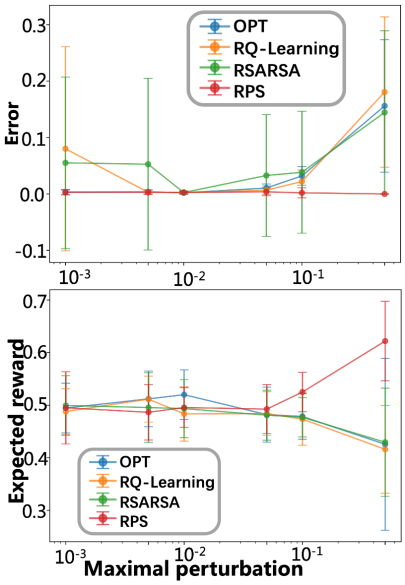

Figure 2: Results on 4 methods. Upper figure: worst perturbation. Below figure: random perturbation.

Assume that the true MDP M^* and M^s differs in the transitions. We assume a perturbation range u for this example. That is, $P^*(s_2|s_0, a_0) \in [0, u)$ and $P^*(s_4|s_0, a_1) \in [0, u)$. Now if $u > 0.02$, the robust policy for the agent is to choose a_1 at s_0. It can be seen that methods concentrating on M^s can hardly identify this robustness issue since they do not pay attention to s_4 whose reaching probability in M^s is 0.

We aim to use this simple case to show that: (1) the optimal policy of M^s may be a bad policy for M^*; (2) the online training process for RMDP can be very inefficient. Thus we test 4 methods: (1) OPT: solving the optimal policy of M^s; (2) RPS: our robust policy solving method; (3) RQ-learning: the robust version of Q-learning; and (4) RSARSA: the robust version of SARSA. Here OPT and RPS are trained with the generative model and RQ-learning and RSARSA (Roy, Xu, and Pokutta 2017) solves a one-step mini-max optimization. Each method can interact with the training environment for 20000 times. We choose $u \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. We give detailed information about this experiment is given in the appendix[2].

The results under worst-case perturbation are shown in the upper of Fig. 2. Only RPS can always finding the robust solution while other three methods suffer a large error for a large u. OTP suffer a large error because it ignores the effect of the perturbation. Online methods RSARSA and RQ-learning also fail because these online methods cannot reach s_4 at all. They cannot give a proper estimation for s_4 and cannot find the near-optimal robust solution.

We also show the results on random perturbation in the

[2]The full version is available at https://arxiv.org/abs/2203.06587.

Algorithm 1: Solving the robust policy with a Nash Equilibium (NE) solver

Input: The generative model M, the perturbation constraint U and the NE solver $\mathcal{O}$
for (s, a) in $\mathcal{S} \times \mathcal{A}$ **do**
  Query M with (s, a) for N times
  Calculate the transition empirical mean $\bar{P}(s, a)$ and the reward empirical mean $\bar{r}(s, a)$
end for
Get an empirical model $\bar{M} = \langle \mathcal{S}, \mathcal{A}, \bar{P}, \bar{r}, H \rangle$.
Solve the NE $(\widehat{\pi}^*, \widehat{\sigma}^*) = \mathcal{O}(\bar{M}, U(\bar{P}))$
Output: The robust policy $\widehat{\pi}^*$

below of Fig. 2. The random perturbation here means that the difference between the training and testing transitions is sampled uniformly from the perturbation range. It can be seen that for a relatively large perturbation, a robust policy indeed gives higher expected rewards, since it takes the risks into consideration.

Methodology

We now present a method to solve for the robust policy in RMDP. It is hard to solve the problem when the uncertainty estimation process and the robust policy solving process are entangled. We treat them separately by re-formalizing the RMDP as a two-player zero-sum game where the agent is a player maximizing reward and the perturbation is the other player minimizing the reward. We deal with the training estimation error by using sufficient samples to build an proper model for the game. Then we can use a plug-in Nash Equilibrium (NE) solver to solve the game. The NE corresponds to the robust policy and the worst-case perturbation.

The idea of solving a minimax dynamic programming have been used in previous works (Nilim and El Ghaoui 2004; Iyengar 2005). The key advantage of considering a two-player zero-sum game rather than simply a minimax dynamic programming is that we can handle the perturbation in a more general way. By doing so, we can estimate the environment uncertainty with RL methods and solve the robust policy with game-theoretical techniques. This separation can help us to handle problems with more complex environment uncertainty or mismatch constraints. We give more examples in the next section.

In this section, we first re-formulate the RMDP problem as a two-player zero-sum game and then present a method for RMDP with general constraint functions. Finally we give a non-asymptotic sample complexity result for our method under a mild assumption.

RMDP as a Two-player Zero-sum Game

We consider MDP M with a given transition P as well as the corresponding $U(P)$ defined in Eq. (1). We denote σ as the elements in $U(P)$. For convenience, we use $M + \sigma$ to denote a perturbed MDP with transition function $P + \sigma$. The goal to learn a robust policy is to solve a maximin problem

$$\max_{\pi} \min_{\sigma \in U(P)} V^{\pi}_{M+\sigma}(s_0). \tag{5}$$

With the above formulation, we can consider this task as a two-player zero-sum game where player 1 chooses π to maximize $V^{\pi}_{M+\sigma}(s_0)$ while player 2 chooses σ to minimize $V^{\pi}_{M+\sigma}(s_0)$. From this point of view, we define $\mathcal{V}$ and $\mathcal{Q}$ values to replace the V and Q values as

$$\mathcal{V}^{\pi,\sigma}_{M}(s) := V^{\pi}_{M+\sigma}(s),\ \mathcal{Q}^{\pi,\sigma}_{M}(s,a) := Q^{\pi}_{M+\sigma}(s,a). \tag{6}$$

Here (π, σ) are the policy pair of the two players and $\mathcal{V}^{\pi,\sigma}_{M}(s)$ represents the expected value for player 1 at state s of depth h. The expected reward for player 2 is accordingly $-\mathcal{V}^{\pi,\sigma}_{M}(s)$.

We further use $\sigma(s, a)$ and $\sigma(s'|s, a)$ to denote the perturbations added to $P(s, a)$ and $P(s'|s, a)$, respectively. Then the Bellman Equation for this game can be shown as:

$$\mathcal{V}^{\pi,\sigma}_{M}(s) = r(s, \pi(s)) + \sum_{s'} \Big(P(s'|s, \pi(s)) + \sigma(s'|s, \pi(s)) \Big) \mathcal{V}^{\pi,\sigma}_{M}(s').$$

Solving problem (5) is equivalent to solving the Nash Equilibrium (NE) of the game. More specifically, the NE solution is corresponding to the robust policy and the worst-case perturbation. Note that there might be multiple NEs for one game. In two-player zero-sum games, all the NEs have the same reward values. Thus any NE can be considered as a solution of the original problem.

Now we turn to solve the NE of the game. We assume that we have access to an NE solver $\mathcal{O}$ which maps the MDP M and perturbation set $U(P)$ to a policy pair $(\pi', \sigma') = \mathcal{O}(M, U(P))$ that satisfies

$$\pi' = \arg\max_{\pi} \mathcal{V}^{\pi,\sigma'}_{M}(s_0),\ \sigma' = \arg\min_{\sigma \in U(P)} \mathcal{V}^{\pi',\sigma}_{M}(s_0).$$

Then the policy pair $(\pi^*, \sigma^*) = \mathcal{O}(M^s, U(P^s))$ is exactly an NE for our learning goal of problem (5) and π^* is exactly the robust policy we aim to solve.

Method for General Constraints

Based on the above formulation, our method is simple and intuitive. We consider our setting that a generative model $M^s = \langle \mathcal{S}, \mathcal{A}, P^s, r, H \rangle$ is given for training. For each state-action pair (s, a), we query the generative model for N times and gain corresponding samples for next steps and rewards, denoted as $\{s_t, r_t\}_{t=1}^{N}$. The number N is calculated based on the target ϵ and δ, as we will give in Sec.. Since we assume the reward to be deterministic and we have visited each state-action pair for N times, we already know the reward r. Here use $\mathbb{I}[\cdot]$ as the indicator function. With these N samples, we estimate the transition function at (s, a) with

$$\widehat{P}(s'|s, a) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{I}[s_t = s']. \tag{7}$$

We construct an empirical RMDP $\widehat{M} = \langle \mathcal{S}, \mathcal{A}, \widehat{P}, r, H \rangle$ and calculate its perturbation set $U(\widehat{P})$. We directly construct a two-player zero-sum game with $\widehat{M}$ and $U(\widehat{P})$, as in Sec. . We apply the NE solver $\mathcal{O}$ to this game to output the policy pair $(\widehat{\pi}^*, \widehat{\sigma}^*) = \mathcal{O}(\widehat{M}, U(\widehat{P}))$ such that:

$$\widehat{\pi}^* = \arg\max_{\pi} \mathcal{V}^{\pi,\widehat{\sigma}^*}_{\widehat{M}}(s_0),\ \widehat{\sigma}^* = \arg\min_{\sigma \in U(\widehat{P})} \mathcal{V}^{\widehat{\pi}^*,\sigma}_{\widehat{M}}(s_0). \tag{8}$$

For convenience, here we denote $\widehat{\mathcal{V}}^*(s) = \mathcal{V}_{\widehat{M}}^{\widehat{\pi}^*,\widehat{\sigma}^*}(s)$, $\widehat{\mathcal{Q}}^*(s,a) = \mathcal{Q}_{\widehat{M}}^{\widehat{\pi}^*,\widehat{\sigma}^*}(s,a)$ for $s \in \mathcal{S}_h$.

Notice that $(\widehat{\pi}^*, \widehat{\sigma}^*)$ is solved in the constraint $\mathcal{C}(\widehat{M})$, rather than $\mathcal{C}(M^s)$. In the next section, we will show that under some mild assumptions on $\mathcal{C}$, with a polynomial number of samples, $\widehat{M}$ can be a good approximation for M^s and our solution $\widehat{\pi}^*$ is guaranteed to be a near-optimal policy.

Theoretical Analysis

Our method above uses a plug-in NE solver to solve the empirical RMDP. Besides the estimation error caused by the sampling, the policy $\widehat{\pi}^*$ is calculated from the perturbation $U(\widehat{P})$ rather than $U(P^s)$, which causes another error. Intuitively, for a large N, $\widehat{M}$ would be sufficiently close to M^s. Thus the gaps between $\widehat{P}$ and P^s would be bounded. If the error caused by the difference between $U(\widehat{P})$ and $U(P^s)$ can also be bounded, we can show that $\widehat{\pi}^*$ is a near-optimal policy. Here we make mild assumptions for $\mathcal{C}$ under which we can gain a near-optimal $\widehat{\pi}^*$ from a polynomial number of samples from the generative model.

To show that the policy $\widehat{\pi}^*$ learned by the plug-in NE solver is near optimal, we need to show that the error term $Err(\widehat{\pi}^*)$ defined in Eq. (3) can be bounded by a small value with a high probability. In general, we cannot guarantee $\widehat{\pi}^*$ to be the near-optimal robust policy even when N is large, because the difference between $U(P^s)$ and $U(\widehat{P})$ can cause a large error. We add a mild assumption for the constraint $\mathcal{C}$ and then we can give sample complexity analysis.

Assumption 1 (Lipschitz condition for the perturbation)**.** *For two MDPs $M = \langle \mathcal{S}, \mathcal{A}, P, r, H\rangle$ and $M' = \langle \mathcal{S}, \mathcal{A}, P', r, H\rangle$, the constraint $\mathcal{C}$ and its corresponding perturbation U satisfy that for any $\sigma \in U(P)$ and the projection of σ onto $U(P')$,*

$$\sigma' := \mathop{\arg\min}_{\sigma'' \in U(P')} \left(\max_{(s,a)} ||\sigma''(s,a) - \sigma(s,a)||_1 \right),$$

there exits a constant λ such that for any (s,a),

$$||\sigma(s,a) - \sigma'(s,a)||_1 \le \lambda ||P(s,a) - P'(s,a)||_1. \tag{9}$$

This assumption indicates that the distance of $\sigma' \in U(P')$ and $\sigma \in U(P)$ satisfies a Lipschitz condition, where σ' is the closest perturbation in $U(P')$ to σ. This is a relatively general assumption for constraints. For example, the total variance distance, where the total variance of the perturbation is bounded, satisfies this assumption by choosing $\lambda = 2$.

For constraints satisfying Assumption 1, our learned policy is guaranteed to be near-optimal, as shown by the following theorem.

Theorem 1. *For a given (ϵ, δ), where $\epsilon \in (0, H)$ and $\delta \in (0,1)$, and a constraint $\mathcal{C}$ satisfying Assumption 1, if*

$$N \ge \frac{8(1+\lambda)^2 H^4 D \ln(2/\delta')}{\epsilon^2}, \tag{10}$$

then with a probability no less than $1-\delta$, we have

$$Err(\widehat{\pi}^*) \le \epsilon. \tag{11}$$

The complete proof for this theorem is given in the appendix. Below we give a brief proof overview.

Proof. The proof process can be decomposed into 3 steps. In the first step we decompose $Err(\widehat{\pi}^*)$ into 2 terms, and then we provide upper bound for the two terms correspondingly.

We first define four auxiliary policies as

$$\sigma^* = \mathop{\arg\min}_{\sigma \in U(P^s)} \mathcal{V}_{M^s}^{\pi^*,\sigma}(s_0),\ \widetilde{\sigma}^* = \mathop{\arg\min}_{\sigma \in U(P^s)} (\max_{(s,a)} ||\widehat{\sigma}^* - \sigma||_1),$$

$$\sigma' = \mathop{\arg\min}_{\sigma \in U(P^s)} \mathcal{V}_{M^s}^{\widehat{\pi}^*,\sigma}(s_0),\ \widetilde{\sigma}' = \mathop{\arg\min}_{\sigma \in U(\widehat{P})} (\max_{(s,a)} ||\sigma' - \sigma||_1).$$

The left two represent the best responses to π^* and $\widehat{\pi}^*$ respectively under $U(P^s)$. $\widetilde{\sigma}^*$ is the projection of $\widehat{\sigma}^*$ (recall the definition in Eq. (8), and notice that $\widehat{\sigma}^* \in U(\widehat{P})$.) onto $U(P^s)$. The last $\widetilde{\sigma}'$ is the projection of σ' onto $U(\widehat{P})$.

Now we sketch the proof in 3 steps.

Step 1. We first provide an upper bound for $Err(\widehat{\pi}^*)$ with the following lemma.

Lemma 1.

$$Err(\widehat{\pi}^*) \le e_1 + e_2, \tag{12}$$

where

$$e_1 := \mathcal{V}_{M^s}^{\pi^*,\widetilde{\sigma}^*}(s_0) - \mathcal{V}_{\widehat{M}}^{\pi^*,\widehat{\sigma}^*}(s_0),\ e_2 := \mathcal{V}_{\widehat{M}}^{\widehat{\pi}^*,\widetilde{\sigma}'}(s_0) - \mathcal{V}_{M^s}^{\widehat{\pi}^*,\sigma'}(s_0).$$

Here e_1 measures the error from transferring $\widehat{\sigma}^*$ on $\widehat{M}$ to $\widetilde{\sigma}^*$ on M^s given fixed π^*. Similarly, e_2 measures the error under $\widehat{\pi}^*$. We then bound the two errors separately.

Step 2. For e_1, notice that the difference between the two $\mathcal{V}$ values is caused by the difference of transitions and perturbations. We can decompose e_1 into state-action pair-wise error terms (refer to Appendix for more details). For clarity, we use $\mathcal{V}_{\widehat{M},h}^{\pi^*,\widehat{\sigma}^*}$ to denote the vector composed of all $\mathcal{V}_{\widehat{M}}^{\pi^*,\widehat{\sigma}^*}(s)$ where $s \in \mathcal{S}_h$. Then we have

$$\begin{aligned}
&\mathcal{V}_{M^s}^{\pi^*,\widetilde{\sigma}^*}(s_0) - \mathcal{V}_{\widehat{M}}^{\pi^*,\widehat{\sigma}^*}(s_0) \\
=&\sum_{h=1}^{H} \sum_{s \in \mathcal{S}_h} \sum_{a} \xi_{M^s}^{\pi^*,\widetilde{\sigma}^*}(s,a) \left((P^s(s,a) - \widehat{P}(s,a))^\top \mathcal{V}_{\widehat{M},h+1}^{\pi^*,\widehat{\sigma}^*} \right) \qquad (13)\\
+&\sum_{h=1}^{H} \sum_{s \in \mathcal{S}_h} \sum_{a} \xi_{M^s}^{\pi^*,\widetilde{\sigma}^*}(s,a) \left((\widetilde{\sigma}^*(s,a) - \widehat{\sigma}^*(s,a))^\top \mathcal{V}_{\widehat{M},h+1}^{\pi^*,\widehat{\sigma}^*} \right). \qquad (14)
\end{aligned}$$

where $\xi_M^{\pi,\sigma}(s,a)$ is defined as the probability of reaching (s,a) on MDP M following policy π and perturbation σ.

Here the first summation term characterizes the estimation error caused by the samples, and the second characterizes the error caused by using an improper perturbation set.

Consider $(P^s(s,a) - \widehat{P}(s,a))^\top \mathcal{V}_{\widehat{M},h+1}^{\pi^*,\widehat{\sigma}^*}$ in Eq. (13). For general constraint $\mathcal{C}$, $\widehat{\sigma}^*$ can be correlated with $\widehat{P}$. Therefore, $\widehat{P}$ and $\mathcal{V}_{\widehat{M},h+1}^{\pi^*,\widehat{\sigma}^*}$ are not independent random variables. According to the Azuma's inequality on ℓ_1-norm, we have that with a probability at least $1-\delta'$,

$$||P^s(s,a) - \widehat{P}(s,a)||_1 \le \alpha(N), \tag{15}$$

where we define

$$\alpha(N) = \sqrt{2D\ln(2/\delta')/N}. \quad (16)$$

Recall our definition of $D \geq |\mathcal{S}_h|$ for all $h \in [H]$. This property holds for all (s,a) pairs.

Using the Cauchy-Schwarz inequality, we have that

$$\begin{aligned}&(P^s(s,a) - \widehat{P}(s,a))^\top \mathcal{V}^{\pi^*,\widehat{\sigma}^*}_{\widehat{M},h+1} \\ \leq &||P^s(s,a) - \widehat{P}(s,a)||_1 ||\mathcal{V}^{\pi^*,\widehat{\sigma}^*}_{\widehat{M},h+1}||_\infty \leq H\alpha(N). \end{aligned} \quad (17)$$

For the $(\widetilde{\sigma}^*(s,a) - \widehat{\sigma}^*(s,a))^\top \mathcal{V}^{\pi^*,\widehat{\sigma}^*}_{\widehat{M},h+1}$ in Eq. (14), we use the Cauchy–Schwarz inequality and Assumption 1 to get

$$(\widetilde{\sigma}^*(s,a) - \widehat{\sigma}^*(s,a))^\top \mathcal{V}^{\pi^*,\widehat{\sigma}^*}_{\widehat{M},h+1} \leq \lambda H\alpha(N). \quad (18)$$

Recall our definition for reaching probability $\xi^{\pi,\sigma}_M(s,a)$ and we have that $\sum_{s\in\mathcal{S}_h, a\in\mathcal{A}} \xi^{\pi,\sigma}_M(s,a) = H$.

Notice that α is a decreasing function with respect to N, and converges to 0 as N goes to infinity. Then with

$$N \geq \alpha^{-1}(\epsilon/(2(1+\lambda)H^2)) = \frac{8(1+\lambda)^2H^4D\ln(2/\delta')}{\epsilon^2}, \quad (19)$$

we have that $e_1 \leq \epsilon/2$.

Step 3. Using similar analysis techniques (refer to Appendix) and N in Eq. (19), we also have $e_2 \leq \epsilon/2$. Finally, we choose $\delta' = \delta/(4SA)$ and use the union bound to get that with a probability no less than $1-\delta$, $Err(\widehat{\pi}^*) \leq \epsilon$. □

By roughly considering D as S, this theorem shows that with a sample complexity of $\widetilde{O}(S^2AH^4(1+\lambda)^2/\epsilon^2)$, our method with a plug-in NE solver can find an ϵ-optimal policy with a high probability. Further if our NE solver returns an approximate NE with a value error ϵ', then we can simply replace ϵ with $\epsilon - \epsilon'$ to calculate N.

Cases and Extensions

In this section, we give some examples for RMDPs. Finally we give a further extension which is also suitable to apply our game-theoretical framework.

Pair-Wise Constraint (PWC)

Here we consider a specific kind of constraints, where the constraint $\mathcal{C}$ has independent effect on each state-action pair. We call this as the *Pair-Wise Constraint* (PWC). Many practical problems provide constraints directly on state-action pairs, such as Total Variation distance bound on transitions. These constraints can be included in the PWC.

For PWC problems, we can implement a simple NE solver based on value back-propagation, which has a quite similar form to methods in previous works (Nilim and El Ghaoui 2004; Iyengar 2005). Detailed definition for PWC and its NE solver are given in Appendix. Here we only give its theoretical result.

Theorem 2. *For a given (ϵ,δ), where $\epsilon \in (0,H)$ and $\delta \in (0,1)$, and a PWC $\mathcal{C}$ satisfying Assumption 1, if*

$$N \geq \frac{8H^4\ln(8SA/\delta)}{\epsilon^2}(2+\lambda\sqrt{D})^2, \quad (20)$$

then with a probability no less than $1-\delta$,

$$Err(\widehat{\pi}^*) \leq \epsilon. \quad (21)$$

The detail of the proof is given in Appendix. This theorem shows that compared with Theorem 1, the sample complexity of PWCs for model estimation can be improved from $\widetilde{O}(DSAH^4/\epsilon^2)$ to $\widetilde{O}(SAH^4/\epsilon^2)$, while the bound for perturbation set error still suffers a scale of $D\lambda^2$.

We also give more results for specific PWC cases. For RMDP with fixed perturbation sets, where the perturbation is independent of the environment parameters, we have $Err(\widehat{\pi}^*) \leq \epsilon$ with a probability no less than $1-\delta$, with $N \geq \frac{8H^4\ln(8SA/\delta)}{\epsilon^2}$. Notice that this sample complexity is comparable to the results of solving MDPs without perturbation, which has an order of $\widetilde{O}(SAH^3\min(H,S)/\epsilon^2)$ (See Sec. C.2 of Cui and Yang (2020)). In other words, if our estimation of $\widehat{P}$ would not cause a perturbation difference, the complexity for solving a RMDP is comparable to that of solving an MDP.

Further we consider another kind of PWC constraint, the Total Variation Distance Constraint (TVDC). For a fixed value u, for each state-action pair (s,a) and any probability $p \in \Delta_{sa}$, the TVDC is defined as

$$\mathcal{C}_{TVD}(p;u) := \{p' \in \Delta_{sa} : ||p'-p||_{TV} \leq u\}. \quad (22)$$

We also aim to ensure $Err(\widehat{\pi}^*) \leq \epsilon$ with a probability no less than $1-\delta$. If $u \leq \epsilon/(16H^2)$, then we can choose $N \geq \frac{128H^4\ln(8SA/\delta)}{\epsilon^2}$; otherwise we choose $N \geq \max\left(\frac{16H^4\ln(8SA/\delta)}{\epsilon^2}(\sqrt{2}+6\sqrt{uD})^2, \frac{49D\ln(8SA/\delta)}{27u}\right)$.

The complete theorem and its proof are given in Appendix. We can see that for a small u, we can nearly consider the TVDC as a fixed perturbation constraint and it has the sample complexity order comparable to that of solving a MDP without perturbation. For $u > \epsilon/(16H^2)$, the perturbation set difference would cause a scale of $1+uD$ on the sample complexity. That is, the larger u is, the larger the sample complexity is needed to guarantee the $Err(\widehat{\pi}^*)$. Finally, there is a higher order term of $49D\ln(8SA/\delta)/(27u) \leq 49*16H^2D\ln(8SA/\delta)/(27\epsilon)$.

Homogeneous Perturbation

PWC problems can be solved by minimax dynamic programming, which is quite similar to previous methods. Here we give a simple example to show that our game-theoretical framework is able to handle more flexible constraints. Here we consider a homogeneous perturbation (HP), where the perturbation on all state-action pairs should be the same. Here we simply assume that Δ_{sa} has the same dimension for all (s,a) and perturbation σ from a fixed perturbation set U is added on all state-action pairs simultaneously. For HP, we cannot choose a worst-case perturbation for each (s,a), and we need to consider the MDP parameters as a whole.

This problem cannot be solved by a directly minimax optimization on state-action pairs. From our game-theoretical point of view, our two-player zero-sum game defined in Sec. reduce to a two-player zero-sum extensive game with imperfect information (TZEGI) (Refer Sec. 3.7 of Nisan et al. (2007) and Def 1 of Zinkevich et al. (2007)). In TZEGI, players make decisions on infosets, which includes all the states that the player cannot distinguish. Here we consider

the perturbation play takes action the infoset which includes all state-action pairs. Therefore we only need to find an NE solver for TZEGI. Fortunately Zinkevich et al. (2007) provide us an approximate NE solver, the counterfactual regret minimization (CFR)[3]. We can use CFR as the solver to return an $\epsilon/4$-NE and choose $N = O(H^4 D \ln(1/\delta)/\epsilon^2)$.

Hence our game-theoretical framework indeed provide us a way to handle RMDP with more general constraints. More detailed discussion for this is given in Appendix.

Extension to Robust Partial Observable MDP

Our method is even able to extend to robust Partial Observable MDP (RPOMDP) problems, where the agent is given observations instead of states.

Consider an RPOMDP problem with the training environment parameters give. The key to solve an RPOMDP is to formalize this problem as a TZEGI, where the agent's infosets includes all the histories it cannot distinguish now. Again we apply the CFR algorithm to find the robust policy for this task. We give detailed discussion in Appendix.

Discussions

Here we give some discussions for our results.

Rectangularity Assumption (RA): many previous works (Yang, Zhang, and Zhang 2021) consider RA, which is also a PWC and is appliable with Thm. 2. The standard robust dynamic programming technique is suitable for RA, but cannot be used under our weak Lipschitz assumption.

Lower bound: Azar, Munos, and Kappen (2013) give the lower bound for solving an episodic MDP with a generative model of order $O(SAH^3 \ln(SA/\delta)/\epsilon^2)$. This can be considered as a lower bound for RMDP. However, it is not clear whether the lower bound can be improved according to the property of the constraints. For the fixed perturbation constraint and the TVDC with small u, our method matches this lower bound except an extra H. For the general constraint with a Lipschitz constant λ, our upper bound matches the lower bound except an extra scale of $HD(1+\lambda)^2$.

Online setting: In the online setting, the agent interacts with the environment to gain trajectories and uses the trajectories to learn policy. The lower bound for online and generative model settings are the same (Azar, Munos, and Kappen 2013). However, for the RMDP problem in this work, the online methods might suffer a sample complexity growing exponentially with H or even infinity. The key challenge is that the agent in the online problem can only reach states according to P^s. For example, consider the case that the reaching probability to a state s under MDP M^s and any policy π is 0, while it can be reached under the true environment M^*. Then the agent can never know the reward at and after s, and this might cause a large error. It is still not clear whether we can solve this issue for the online setting.

Comparison with MDP without perturbations: Compared to current results of $\widetilde{O}(SAH^3 \min(H,S)/\epsilon^2)$ in Cui and Yang (2020) for finite-horizon MDPs without perturbation, our result in Theorem 1 suffers an extra $D(1+\lambda)^2$ term. The term $(1+\lambda)^2$ comes from the Lipschitz condition of $\mathcal{C}$, and the D appears as we require transitions for all state-action pair (s,a) to be tight enough. The appearance of D is similar to the conclusion of the recently studied reward-free reinforcement learning setting (e.g. Jin et al. 2020; Zhang, Du, and Ji 2020).

[3]Notice that CFR usually returns a random policy.

Related Work

Many existing works concentrate on the training and testing mismatch problem on MDPs. A straightforward way is to connect the two MDPs and transfer knowledge from M^s to M^*. Rusu et al. (2017) apply transfer learning to transfer features for testing. Jiang (2018) theoretically gives some results on how to repair a mismatched training MDP.

As for the problems where the testing environment is not accessible, there are also various works focusing on solving the robust policies via maximin methods. Dupačová (1987) discusses the minimax approaches for stochastic problems and Shapiro and Kleywegt (2002) give a Bayesian approach. Nilim and El Ghaoui (2004) and Iyengar (2005) propose a general framework for the maximin solutions for RMDPs. However, they require full knowledge of M^s and the perturbations. Wiesemann, Kuhn, and Rustem (2013) concentrate on RMDP under RA and Puggelli et al. (2013) focus on convec uncertainties. Recent work proposes a model-free method for RMDPs (Roy, Xu, and Pokutta 2017) and the extension to MDPs with functional approximations is also considered (Tamar, Mannor, and Xu 2014; Panaganti and Kalathil 2020). Derman et al. (2020) propose a Bayesian method for RMDPs. Although they consider the problems where the interactions with M^s are needed, they lack non-asymptotic analysis for the sample complexity. Some further work (Cubuktepe et al. 2021) considers RPOMDPs under certain assumptions.

As for the solving an MDP with a generative model, plenty of works (Kearns and Singh 1999; Kakade 2003; Azar, Munos, and Kappen 2012, 2013; Cui and Yang 2020) have been done for the sample complexity to find an optimal policy. They mostly concentrate on infinite-horizon MDP with a discount factor γ and currently have reached the lower bound of $\widetilde{O}(SA/((1-\gamma)^3\epsilon^2)$. However, their results cannot be applied to RMDPs directly since the perturbation makes the environment for the agent no longer stable.

Conclusion

We focus on the policy learning for Robust Markov Decision Process where the training and testing MDPs are mismatched. For general constraints on the perturbation, we solve RMDP problems by re-formalizing the problem as finding an NE of a two-player zero-sum game and applying a plug-in NE solver. For a constraint satisfying the Lipschitz condition with a constant λ, our method has a sample complexity of $\widetilde{O}(DSAH^4(1+\lambda)^2/\epsilon^2)$. Our method can solve constraints like PWC and HP and is able to extend to solve problems like RPOMDPs.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.s 2020AAA0106000, 2020AAA0104304, 2020AAA0106302), NSFC Projects (Nos. U19A2081, 62061136001, 62076147, U19B2034, U1811461), Beijing NSF Project (No. JQ19016), Tsinghua-Huawei Joint Research Program, and Tsinghua Institute for Guo Qiang.

References

Azar, M. G.; Munos, R.; and Kappen, B. 2012. On the sample complexity of reinforcement learning with a generative model. *arXiv preprint arXiv:1206.6461*.

Azar, M. G.; Munos, R.; and Kappen, H. J. 2013. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3): 325–349.

Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q.; Hanasusanto, G. A.; and Kuhn, D. 2013. Robust data-driven dynamic Programming. *Advances in Neural Information Processing Systems*, 827–835.

Comair, G. F.; Mckinney, D.; Maidment, D. R.; Espinoza, G.; Sangiredy, H.; Fayad, A.; and Salas, F. R. 2014. Hydrology of the Jordan River Basin: A GIS-based system to better guide water resources management and decision making. *Water Resources Management An International Journal Published for the European Water Resources Association*, 28(4): 933–946.

Cubuktepe, M.; Jansen, N.; Junges, S.; Marandi, A.; Suilen, M.; and Topcu, U. 2021. Robust finite-state controllers for uncertain POMDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11792–11800.

Cui, Q.; and Yang, L. F. 2020. Is plug-in solver sample-efficient for feature-based reinforcement learning? *arXiv preprint arXiv:2010.05673*.

Delage, E.; and Mannor, S. 2010. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58: 203–213.

Derman, E.; Mankowitz, D.; Mann, T.; and Mannor, S. 2020. A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*, 648–658. PMLR.

Dupačová, J. 1987. The minimax approach to stochastic programming and an illustrative application. *Stochastics: An International Journal of Probability and Stochastic Processes*, 20(1): 73–88.

Ghavamzadeh, M.; Petrik, M.; and Chow, Y. 2016. Safe policy improvement by minimizing robust baseline regret. *Advances in Neural Information Processing Systems*, 29: 2298–2306.

Iyengar, G. N. 2005. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280.

Jiang, N. 2018. PAC reinforcement learning with an imperfect model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jin, C.; Krishnamurthy, A.; Simchowitz, M.; and Yu, T. 2020. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, 4870–4879. PMLR.

Kakade, S. M. 2003. *On the sample complexity of reinforcement learning*. Ph.D. thesis, UCL (University College London).

Kearns, M.; and Singh, S. 1999. Finite-sample convergence rates for Q-learning and indirect algorithms. *Advances in Neural Information Processing Systems*, 996–1002.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Nilim, A.; and El Ghaoui, L. 2004. Robustness in Markov decision problems with uncertain transition matrices. In *Advances in Neural Information Processing Systems*, 839–846.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic Game Theory*. Cambridge University Press.

Panaganti, K.; and Kalathil, D. 2020. Model-free robust reinforcement learning with linear function approximation. *arXiv e-prints*, arXiv–2006.

Puggelli, A.; Li, W.; Sangiovanni-Vincentelli, A. L.; and Seshia, S. A. 2013. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *International Conference on Computer Aided Verification*, 527–542. Springer.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rahimian, H.; and Mehrotra, S. 2019. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*.

Rasheed, A.; San, O.; and Kvamsdal, T. 2020. Digital twin: values, challenges and enablers from a modeling perspective. *IEEE Access*, PP(99): 1–1.

Roy, A.; Xu, H.; and Pokutta, S. 2017. Reinforcement learning under model mismatch. In *Advances in Neural Information Processing Systems*, 3043–3052.

Rusu, A. A.; Večerík, M.; Rothörl, T.; Heess, N.; Pascanu, R.; and Hadsell, R. 2017. Sim-to-real robot learning from pixels with progressive nets. In *Conference on Robot Learning*, 262–270. PMLR.

Shapiro, A.; and Kleywegt, A. 2002. Minimax analysis of stochastic problems. *Optimization Methods and Software*, 17(3): 523–542.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676): 354–359.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling up robust MDPs using function approximation. In *International Conference on Machine Learning*, 181–189. PMLR.

Wiesemann, W.; Kuhn, D.; and Rustem, B. 2013. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1): 153–183.

Yang, W.; Zhang, L.; and Zhang, Z. 2021. Towards Theoretical Understandings of Robust Markov Decision Processes: Sample Complexity and Asymptotics. *arXiv preprint arXiv:2105.03863*.

Zhang, Z.; Du, S. S.; and Ji, X. 2020. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20: 1729–1736.