Scaling Up Robust MDPs using Function Approximation

Aviv Tamar AVIVT@TX.TECHNION.AC.IL
Electrical Engineering Department, The Technion - Israel Institute of Technology, Haifa 32000, Israel

Shie Mannor SHIE@EE.TECHNION.AC.IL
Electrical Engineering Department, The Technion - Israel Institute of Technology, Haifa 32000, Israel

Huan Xu MPEXUH@NUS.EDU.SG
Mechanical Engineering Department, National University of Singapore, Singapore 117575, Singapore

Abstract

We consider *large-scale* Markov decision processes (MDPs) with parameter uncertainty, under the robust MDP paradigm. Previous studies showed that robust MDPs, based on a minimax approach to handling uncertainty, can be solved using dynamic programming for *small to medium sized* problems. However, due to the "curse of dimensionality", MDPs that model real-life problems are typically prohibitively large for such approaches. In this work we employ a reinforcement learning approach to tackle this planning problem: we develop a *robust approximate dynamic programming* method based on a projected fixed point equation to approximately solve large scale robust MDPs. We show that the proposed method provably succeeds under certain technical conditions, and demonstrate its effectiveness through simulation of an option pricing problem. To the best of our knowledge, this is the first attempt to scale up the robust MDP paradigm.

1. Introduction

Markov decision processes (MDPs) are standard models for sequential decision making problems in stochastic dynamic environments (Puterman, 1994; Bertsekas & Tsitsiklis, 1996). Given the parameters, namely, transition probability and reward, the strategy that achieves maximal expected accumulated reward is considered optimal. However, in practice, these parameters are typically estimated from noisy data, or even worse, they may change during the execution of a policy. It is thus not surprising that the actual performance of the chosen strategy can significantly degrade from the model's prediction due to such *parameter uncertainty* – the deviation of the model parameters from the true ones (see experiments in Mannor et al. 2007).

To mitigate performance deviation due to parameter uncertainty, the robust MDP framework (Iyengar, 2005; Nilim & El Ghaoui, 2005; Bagnell et al., 2001) is now a common method. In this context, it is assumed that the *uncertain* parameters can be any member of a known set (termed the "uncertainty set"), and solutions are ranked based on their performance under the (respective) worst parameter realizations. Under mild technical conditions, the optimal solution of a robust MDP can be obtained using dynamic programming, at least for small to medium sized MDPs.

This paper considers planning in large robust MDPs, a setting largely untouched in literature. It is widely known that, due to the "curse of dimensionality", practical problems modeled as MDPs often have prohibitively large state-spaces, under which dynamic programming becomes intractable. Many approximation schemes have been proposed to alleviate the curse of dimensionality of large scale MDPs, among them approximate dynamic programming (ADP) is a popular approach (Powell, 2011). ADP considers approximations of the optimal value function, for example, as a linear functional of some features of the state, that can be solved efficiently using a sampling based approach. Of course, selecting good features is an art by itself. However, ADP has been used successfully in large-scale problems with hundreds of state dimensions (Powell, 2011). Inspired by the empirical success of ADP, we adapt it to the robust MDP setting, and develop and analyze methods that handle large scale robust MDPs. From a high level, we indeed solve a planning problem via a reinforcement learning (RL; Sutton & Barto 1998) approach: while the robust MDP model, the parameters, and the uncertainty sets are all known, and hence the optimal solution is well defined, we still use an RL approach to approximately find the solution

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

due to the scale of the problem.

Our specific contributions are a framework for approximate solution of large-scale robust MDPs; algorithms for approximate robust policy evaluation and policy improvement, with convergence proofs and error bounds; and an application of our framework to an option trading domain.

2. Background

We describe our problem formulation and some preliminaries from robust MDPs and ADP.

2.1. Robust Markov Decision Processes

For a discrete set $\mathcal{B}$, let $\mathcal{M}(\mathcal{B})$ denote the set of probability measures on $\mathcal{B}$, and let $|\mathcal{B}|$ denote its cardinality. A Markov Decision Process (MDP; Puterman 1994) is a tuple $\{\mathcal{X}, \mathcal{Z}, \mathcal{U}, P, r, \gamma\}$ where $\mathcal{X}$ is a finite set of states, $\mathcal{Z}$ is a (possibly empty) set of absorbing terminal states, and $\mathcal{U}$ is a finite set of actions. Also, $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is a deterministic and bounded reward function, γ is a discount factor, and $P : \mathcal{X} \times \mathcal{U} \to \mathcal{M}(\mathcal{X} \cup \mathcal{Z})$ denotes the probability distribution of next states, given the current state and action. We assume zero reward at terminal states.

A stationary policy $\pi : \mathcal{X} \to \mathcal{M}(\mathcal{U})$ maps each state to a probability distribution over the actions. The value of a state x under policy π and state transition model P is denoted $V^{\pi,P}(x)$ and represents the expected sum of discounted returns when starting from that state and executing π,

$$V^{\pi,P}(x) = \mathbb{E}^{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \middle| x_0 = x\right],$$

where $\mathbb{E}^{\pi,P}$ denotes expectation w.r.t. the state-action distribution induced by the transitions P and the policy π. Note that for any terminal state $z \in \mathcal{Z}$ and all π and P we have $V^{\pi,P}(z) = 0$.

Typically in MDPs, one is interested in finding a policy that maximizes the value of certain (or all) states. When the state space is small enough, and all the parameters are known, efficient methods exist (Puterman, 1994). In practice, however, the state transition probabilities may not be exactly known. A widely-applied approach in this setting is the Robust MDP (RMDP; Nilim & El Ghaoui 2005; Iyengar 2005, also termed Ambiguous MDP). In this framework, the unknown transition probabilities are assumed to lie in some *known* uncertainty set. Such a set may be obtained, for example, from statistical confidence intervals when the transition probabilities are estimated from data. Mathematically, an RMDP is a tuple $\{\mathcal{X}, \mathcal{Z}, \mathcal{U}, \mathcal{P}, r, \gamma\}$ where $\mathcal{X}, \mathcal{Z}, \mathcal{U}, r$, and γ are as defined for MDPs. The uncertainty set $\mathcal{P}$, where $\mathcal{P}(x, u) \subset \mathcal{M}(\mathcal{X} \cup \mathcal{Z})$, denotes a known uncertainty in the state transitions. Note that this definition implicitly assumes a *rectangularity* of the uncertainty set (Iyengar, 2005). In robust MDPs, one is typically interested in maximizing the *worst case* performance. Formally, we define the robust value function (Iyengar, 2005; Nilim & El Ghaoui, 2005) for a policy π as its worst-case value function

$$V^{\pi}(x) = \inf_{P \in \mathcal{P}} V^{\pi,P}(x),$$

and we seek for the optimal robust value function $V^*(x) = \sup_\pi \left\{\inf_{P \in \mathcal{P}} V^{\pi,P}(x)\right\}$. Iyengar (2005) and Nilim & El Ghaoui (2005) showed that similarly to the regular value function, the robust value function is obtained by a deterministic policy, and satisfies a (robust) Bellman recursion of the form

$$V^*(x) = \sup_{u \in \mathcal{U}} \left\{ r(x, u) + \gamma \inf_{P \in \mathcal{P}} \mathbb{E}^P \left[V^*(x') | x, u\right] \right\},$$

where x' denotes the state following the state x and action u. Thus, in the sequel we shall only consider deterministic policies, and write $\pi(x)$ as the action prescribed by policy π at state x.

Iyengar (2005) proposed a policy iteration algorithm for the robust MDP framework. This algorithm repeatedly improves a policy π by choosing greedy actions with respect to V^π. The key step in this approach is therefore policy evaluation: calculating V^π, which satisfies

$$V^{\pi}(x) = r(x, \pi(x)) + \gamma \inf_{P \in \mathcal{P}} \mathbb{E}^P \left[V^{\pi}(x') | x, \pi(x)\right]. \quad (1)$$

The non-linear equation (1) may be solved for V^π using an iterative method as follows. Let us first write (1) in vector notation. For some x and u we define the operator $\sigma_{\mathcal{P}(x,u)} : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}$ as

$$\sigma_{\mathcal{P}(x,u)} v \doteq \inf \left\{p^\top v : p \in \mathcal{P}(x, u)\right\},$$

where $v \in \mathbb{R}^{|\mathcal{X}|}$ and, slightly abusing notation, we ignore transitions to terminal states in $\mathcal{P}(x, u)$. Also, for some policy π let the operator $\sigma_\pi : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ be defined such that $\{\sigma_\pi v\}(x) \doteq \sigma_{\mathcal{P}(x,\pi(x))} v$. Then (1) may be written as $V^\pi = r^\pi + \gamma \sigma_\pi V^\pi$. Let $T^\pi : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ denote the robust Bellman operator for a fixed policy, defined by

$$T^\pi v \doteq r^\pi + \gamma \sigma_\pi v. \quad (2)$$

We see that V^π is a fixed point of T^π, i.e., $V^\pi = T^\pi V^\pi$. Furthermore, since T^π is known to be a contraction in the sup norm (Iyengar, 2005), V^π may be found by iteratively applying T^π to some vector v.

The robust Bellman operator $T : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$ is defined by

$$Tv(x) \doteq \sup_\pi T^\pi v(x),$$

and was shown to be a contraction (Iyengar, 2005), with V^* as its fixed point.

2.2. Projected Fixed Point Equation Methods

For MDPs, when the state space is large, dynamic programming methods become intractable, and one has to resort to an approximation procedure. A popular approach involves a projection of the value function onto a lower dimensional subspace by means of linear function approximation (Bertsekas & Tsitsiklis, 1996), and solving the solution of a *projected* Bellman equation. We briefly review this approach.

Assume a standard MDP setting without uncertainty, where the Bellman equation (1) for a fixed policy is reduced to $V^\pi(x) = r(x, \pi(x)) + \gamma \mathbb{E}^P V^\pi(x')$, and let T^π_{reg} denote the corresponding fixed policy Bellman operator. When the state space is large, calculating $V^\pi(x)$ for every x is prohibitively computationally expensive, and a lower dimensional approximation of V^π is sought. Consider the linear approximation given by a weighted sum of features

$$\tilde{V}^\pi(x) = \phi(x)^\top w, \quad x \in \mathcal{X},$$

where $\phi(x) \in \mathbb{R}^k$, $k < |\mathcal{X}|$ contains the features of state x and $w \in \mathbb{R}^k$ are the approximation weights. Let $\Phi \in \mathbb{R}^{|\mathcal{X}| \times k}$ denote a matrix with the feature vectors in its rows. We assume that the features are linearly independent, i.e., $rank(\Phi) = k$. A popular approach for finding w is by solving the *projected Bellman equation* (Bertsekas, 2012), given by

$$\tilde{V}^\pi = \Pi T^\pi_{reg} \tilde{V}^\pi, \tag{3}$$

where Π is a projection operator onto the subspace spanned by Φ with respect to a d-weighted Euclidean norm. At this point we only assume that $d \in \mathbb{R}^{|\mathcal{X}|}$ is positive. Since there is no uncertainty, T^π_{reg} is a linear mapping, and Equation (3) may be written in matrix form as follows

$$\Phi^\top D \Phi w = \Phi^\top D r + \gamma \Phi^\top D P^\pi \Phi w, \tag{4}$$

where $D = \text{diag}(d)$, and $P^\pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ is the Markov transition matrix induced by policy π. Given $\Phi^\top D \Phi$, $\Phi^\top D r$, and $\Phi^\top D P^\pi \Phi$, Eq. (4) may be solved for w either by matrix inversion (Boyan, 2002), or iteratively (known as Projected Value Iteration; PVI; Bertsekas 2012)

$$w_{k+1} = \left(\Phi^\top D \Phi\right)^{-1} \left(\Phi^\top D r + \gamma \Phi^\top D P^\pi \Phi w_k\right). \tag{5}$$

When d corresponds to the steady state distribution over states for policy π, the iterative procedure in (5) can be shown to converge using contraction properties of ΠT^π_{reg} (Bertsekas, 2012). For a large state space, the terms in (5) cannot be calculated explicitly. However, the strength of this approach is that these terms may be sampled efficiently, using trajectories from the MDP (Bertsekas, 2012).

Recall that our ultimate goal is policy improvement. For a regular MDP, the policy evaluation procedure described above may be combined with a policy improvement step using Least Squares Policy Iteration (LSPI; Lagoudakis & Parr 2003), which extends policy iteration to the function approximation setting.

3. Robust Policy Evaluation

In this section we propose an extension of ADP to the robust setting. We do this as follows. First, we consider policy evaluation, and extend the projected fixed point equation (3) to the robust case, with the robust T^π operator replacing T^π_{reg}. We discuss the conditions under which this equation has a solution, and how it may be obtained. We then propose a sampling based procedure to solve the equation for large state spaces, and prove its convergence. Finally, in Section 4, we will use our policy evaluation procedure as part of a policy improvement algorithm in the spirit of LSPI (Lagoudakis & Parr, 2003), for obtaining an (approximately) optimal robust policy.

3.1. A Projected Fixed Point Equation

Throughout this section we consider a fixed policy π. For some positive d, let the projection operator Π be defined as above. Consider the following *projected robust Bellman equation* for a fixed policy

$$\tilde{V}^\pi = \Pi T^\pi \tilde{V}^\pi. \tag{6}$$

Note that here, as opposed to (3), T^π is not necessarily linear, and hence it is not clear whether Eq. (6) has a solution at all. We now show that under suitable conditions the operator ΠT^π is a contraction and Equation (6) has a *unique* solution. We consider two different cases, depending on the existence of terminal states $\mathcal{Z}$. Let $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$ represent a given policy, state transition probabilities, and initial state distribution, respectively. We let $\Pr(x_t = j | \hat{\pi}, \hat{P}, \hat{\xi})$ denote the probability that the state at time t is j, given that the states evolve according to a Markov chain with transitions $\hat{P}$, policy $\hat{\pi}$, and initial state distribution $\hat{\xi}$. In the sequel, $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$ will be used to represent the *exploration* policy of the MDP in an offline learning setting. We make the following assumption on $\hat{\pi}$, $\hat{P}$, and $\hat{\xi}$, which also defines the projection weights d.

Assumption 1. *Either $\mathcal{Z} = \emptyset$, and there exists positive numbers d_j such that*

$$d_j = \lim_{t \to \infty} \Pr(x_t = j | x_0 = i, \hat{\pi}, \hat{P}) \quad \forall i, j \in \mathcal{X},$$

or $\mathcal{Z} \neq \emptyset$, and the policy $\hat{\pi}$ is proper *(Bertsekas, 2012), that is, for $\bar{t} = |\mathcal{X}|$*

$$\Pr(x_{\bar{t}} \in \mathcal{Z} | x_0 = i, \hat{\pi}, \hat{P}) > 0 \quad \forall i \in \mathcal{X},$$

and all states have a positive probability of being visited. In this case we let

$$d_j = \sum_{t=0}^{\infty} \Pr(x_t = j | \hat{\pi}, \hat{P}, \hat{\xi}) \quad \forall j \in \mathcal{X}.$$

Put simply, Assumption 1 requires that every state has a positive probability of being visited, and defines d_j as a suitable occupation measure of state j.

The following assumption relates the transitions of the exploration policy and the (uncertain) transitions of the policy under evaluation.

Assumption 2. *There exists* $\beta \in (0,1)$ *such that* $\gamma P(x'|x,\pi(x)) \leq \beta \hat{P}(x'|x,\hat{\pi}(x)), \quad \forall P \in \mathcal{P}, x \in \mathcal{X}, x' \in \mathcal{X}$.

Assumption 2 may appear restrictive, especially when the discount factor γ approaches 1. Unfortunately, it is necessary in the sense that without it ΠT^π is not necessarily a contraction (see supplementary material). We note that a similar difficulty arises in off-policy RL (Bertsekas & Yu, 2009; Sutton et al., 2009), and our Assumption 2 is in fact similar to an assumption of Bertsekas & Yu 2009. Nevertheless, although our algorithms in the sequel are motivated by the contraction property of ΠT^π, we show empirically that our approach works in cases where Assumption 2 is *severely violated*, therefore in practice it is not a serious limitation.

Let $\|\cdot\|_d$ denote the d-weighted Euclidean norm, which is well-defined due to Assumption 1. Our key insight is the following proposition, which shows that under Assumption 2, the robust Bellman operator is a β-contraction in $\|\cdot\|_d$.

Proposition 3. *Let Assumptions 1 and 2 hold. Then* $\|T^\pi y - T^\pi z\|_d \leq \beta \|y - z\|_d$ *for all* $y, z \in \mathbb{R}^{|\mathcal{X}|}$

Proof. Fix $x \in \mathcal{X}$, and assume that $T^\pi y(x) \geq T^\pi z(x)$. Choose some $\epsilon > 0$, and $P_x \in \mathcal{P}$ such that

$$\mathbb{E}^{P_x}\left[z(x') \middle| x, \pi(x)\right] \leq \inf_{P \in \mathcal{P}} \mathbb{E}^{P}\left[z(x') \middle| x, \pi(x)\right] + \epsilon. \quad (7)$$

Also, note that by definition

$$\inf_{P \in \mathcal{P}} \mathbb{E}^{P}\left[y(x') \middle| x, \pi(x)\right] \leq \mathbb{E}^{P_x}\left[y(x') \middle| x, \pi(x)\right]. \quad (8)$$

Now, we have

$$\begin{aligned}
0 &\leq T^\pi y(x) - T^\pi z(x) \\
&\leq (\gamma \mathbb{E}^{P_x}\left[y(x') \middle| x, \pi(x)\right]) - (\gamma \mathbb{E}^{P_x}\left[z(x') \middle| x, \pi(x)\right] - \gamma\epsilon) \\
&= \gamma \mathbb{E}^{P_x}\left[y(x') - z(x') \middle| x, \pi(x)\right] + \gamma\epsilon \\
&\leq \beta \mathbb{E}^{\hat{P}}\left[|y(x') - z(x')| \,\middle|\, x, \hat{\pi}(x)\right] + \gamma\epsilon,
\end{aligned}$$

where the second inequality is by (7) and (8), and the last inequality is by Assumption 2. Conversely, if $T^\pi z(x) \geq T^\pi y(x)$, following the same procedure we obtain $0 \leq T^\pi z(x) - T^\pi y(x) \leq \beta \mathbb{E}^{\hat{P}}\left[|y(x') - z(x')| \,\middle|\, x, \hat{\pi}(x)\right] + \gamma\epsilon$, and we therefore conclude that $|T^\pi y(x) - T^\pi z(x)| \leq \beta \mathbb{E}^{\hat{P}}\left[|y(x') - z(x')| \,\middle|\, x, \hat{\pi}(x)\right] + \gamma\epsilon$. Since ϵ was arbitrary, we have that $|T^\pi y(x) - T^\pi z(x)| \leq \beta \mathbb{E}^{\hat{P}}\left[|y(x') - z(x')| \,\middle|\, x, \hat{\pi}(x)\right]$ for all x, and therefore

$$\|T^\pi y - T^\pi z\|_d \leq \beta \left\| \hat{P}^{\hat{\pi}} |y - z| \right\|_d \leq \beta \|y - z\|_d,$$

where in last equality we used the well-known result that the state transition matrix $\hat{P}^{\hat{\pi}}$ is contracting in the d-weighted Euclidean norm (Bertsekas, 2012). □

The projection operator Π is known to be non-expansive in the d-weighted norm (Bertsekas, 2012). This fact, and Lemma 6.9 of Bertsekas & Tsitsiklis (1996) lead to the following contraction property and error bound for the approximate robust value function $\tilde{V}^\pi$:

Corollary 4. *Let Assumptions 1 and 2 hold. Then the projected robust Bellman operator* ΠT^π *is a* β*-contraction in the* d*-weighted Euclidean norm. Furthermore, Eq.* (6) *has a unique solution, and*

$$\left\| \tilde{V}^\pi - V^\pi \right\|_d \leq \frac{1}{1-\beta} \|\Pi V^\pi - V^\pi\|_d.$$

The contraction property in Corollary 4 also suggests a straightforward procedure for solving Equation (6) which we describe next.

3.2. Robust Projected Value Iteration

Consider the robust equivalent of PVI for solving Eq. (6):

$$\Phi w_{k+1} = \Pi T^\pi \left(\Phi w_k\right). \quad (9)$$

The algorithm (9) may be written explicitly in matrix form (see Bertsekas 2012) as

$$w_{k+1} = \left(\Phi^\top D \Phi\right)^{-1} \left(\Phi^\top D r + \gamma \Phi^\top D \sigma_\pi(\Phi w_k)\right). \quad (10)$$

We refer to the algorithm in (10) as *robust projected value iteration* (RPVI). Note that a matrix inversion approach would not be applicable here, as (10) is not linear due to non-linearity of $\sigma_\pi(\cdot)$.

Corollary 4 guarantees that under Assumptions 1 and 2, the iterates of (9) converge to the fixed point of ΠT^π, and the RPVI algorithm converges to the corresponding weights. We emphasize that Assumption 2 is only a *sufficient* condition for convergence. As we show empirically in Section 5, the algorithm works in cases where Assumption 2 is severely violated, and in fact, we have not encountered convergence issues in any of our experiments. Nevertheless, Assumption 2 does point out where things may go wrong. This is important in practice, especially if the uncertainty set may be controlled to satisfy it. Finally, note that for averager type function approximations (Gordon, 1995), such as non-overlapping grid tiles, kernel smoothing, and k-nearest-neighbor, Π contracts in the sup-norm.

Since T^π also contracts in the sup-norm (Iyengar, 2005), ΠT^π contracts regardless of Assumption 2, and convergence of RPVI is guaranteed.

For a large state space, computing the terms in (10) exactly is intractable. For this case we propose a sampling procedure for estimating these terms, as described next.

3.3. A Sampling Based Approach

When the state space is too large for the terms in Equation (6) to be computed exactly, one may resort to a sampling based procedure. This approach is popular in the RL and ADP literature, and has been used successfully on problems with very large state spaces (Powell, 2011). Here, we describe how it may be applied for the robust MDP setting.

Assume that we have obtained a long trajectory from an MDP with transition probabilities $\hat{P}$, while following policy π. We denote this trajectory by $x_0, u_0, r_0, x_1, u_1, r_1, \ldots, x_N, u_N, r_N$. The terms in (10) may be estimated from the data by[1]

$$\Phi^\top D\Phi \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)\phi(x_t)^\top, \quad \Phi^\top Dr \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)r(x_t,u_t),$$

and

$$\Phi^\top D\sigma_\pi(\Phi w_k) \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t)\sigma_{\mathcal{P}(x_t,u_t)}(\Phi w_k). \quad (11)$$

Using the law of large numbers, it may be proved[2] that these estimates converge with probability 1 to their respective terms in (10) as $N \to \infty$. Together with Corollary 4 we have the following convergence result. The straightforward proof is omitted.

Proposition 5. *Let Assumptions 1 and 2 hold. Consider the RPVI algorithm with the terms in (10) replaced by their sampled counterparts (11). Then as $N \to \infty$ and $k \to \infty$, w_k converges with probability 1 to w^*, and Φw^* is the unique solution of (6).*

3.4. Solving the Inner Problem

In Eq. (11), the calculation of each $\sigma_{\mathcal{P}(x_t,u_t)}(\Phi w_k)$ in the sum requires the solution of the *inner problem*:

$$\inf_{p\in\mathcal{P}(x,u)} \sum_{x\in\mathcal{X}_r(x,u)} p(x)\phi(x)^\top w_k, \quad (12)$$

where $\mathcal{X}_r(x,u)$ denotes the set of reachable states from (x,u) under *all* transitions in the set $\mathcal{P}(x,u)$. Solving Eq. (12) clearly requires a model – i.e., access to the state transitions in $\mathcal{P}(x,u)$. Also, depending on the uncertainty set, it may be computationally demanding. We now discuss *specific* uncertainty sets for which Eq. (12) is tractable.

A natural class of models is constructed from empirical state transitions $x_t \to x_{t+1}$. Let $\hat{p}$ denote the empirical transition frequencies from state x and action u (obtained by, e.g., historical observations of the system), and consider sets on the support of $\hat{p}$ of the form $\mathcal{P}(x,u) = \left\{p : \mathrm{Dist}(p,\hat{p}) \le \epsilon, p^\top \mathbb{1} = 1, p \ge 0\right\}$, where $\mathrm{Dist}(\cdot,\cdot)$ is some distance function and $\epsilon > 0$. The distance function and confidence parameter ϵ are typically related to statistical confidence regions about $\hat{p}$ (Nilim & El Ghaoui, 2005). For the case of the L_1 distance, Strehl & Littman (2005) solve Eq. (12) with complexity $\mathcal{O}(|\hat{p}|\log|\hat{p}|)$. Iyengar (2005) and Nilim & El Ghaoui (2005) propose efficient solutions for the Kullback-Liebler distance, and also for interval and ellipsoidal models. All of these methods scale at least linearly with the number of elements in $\hat{p}$, which in most practical scenarios is small compared to the cardinality of the state space, as it is bounded by the sample size used to create $\hat{p}$. In the case of binary transitions, as in our option pricing example of Section 5, performing the minimization in (12) is trivial.

Nonetheless, some problems may involve very large, or even continuous sets of reachable states. A natural model for these cases is a set of *parametric* distributions. Let $p_\theta(x)$ denote a distribution on $\mathcal{X}$ parameterized by θ. We consider uncertainty sets of the form $\mathcal{P}(x,u) = \{p_\theta : \theta \in \Theta\}$, where Θ is some convex set[3], and our goal is solving

$$\inf_{\theta\in\Theta} \mathbb{E}_{p_\theta}\left[\phi(x)^\top w_k\right]. \quad (13)$$

We assume that we have access to a distribution $\tilde{p}(x)$ such that $p_\theta(x)/\tilde{p}(x)$ is well defined for all $x \in \mathcal{X}$ and $\theta \in \Theta$. Now, observe that (13) may be written as a Stochastic Program (SP): $\inf_{\theta\in\Theta} \mathbb{E}_{\tilde{p}}\left[\frac{p_\theta(x)}{\tilde{p}(x)}\phi(x)^\top w_k\right]$. A standard solution to this SP is via the Sample Average Approximation (SAA; Shapiro & Nemirovski 2005), where N_s i.i.d. samples $x_i \sim \tilde{p}$ are drawn, and the following *deterministic* problem is solved: $\inf_{\theta\in\Theta} \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{p_\theta(x_i)}{\tilde{p}(x_i)}\phi(x_i)^\top w_k$. When the objective of the SP is convex, and under additional technical conditions on $\tilde{p}$, p_θ, and ϕ, efficient solution of (13) is guaranteed[4] (Shapiro & Nemirovski, 2005). An alternative to the SAA is to optimize (13) directly using stochastic mirror descent (Nemirovski et al., 2009), by noting that an unbiased estimate of the gradient may be ob-

[1] These estimates are for the case $\mathcal{Z} = \emptyset$ in Assumption 1. Modifying these estimates for the case $\mathcal{Z} \neq \emptyset$ is straightforward, along the lines of Chapter 7.1 of Bertsekas (2012).

[2] The proof is similar to the case without uncertainty, detailed by Bertsekas (2012).

[3] As a concrete example, consider a Gaussian distribution $p_\theta = \mathcal{N}(\theta, 1)$, where $\Theta = [\theta^-, \theta^+]$, is a confidence interval for the maximum likelihood estimate of θ from historical data.

[4] See the supplementary material for an explicit result.

tained by sampling, using the likelihood ratio trick:

$$\nabla_\theta \mathbb{E}_{p_\theta}\left[\phi(x)^\top w_k\right] = \mathbb{E}_{p_\theta}\left[\nabla_\theta \log p_\theta(x)\phi(x)^\top w_k\right].$$

An in-depth analysis of this approach is deferred to the full version of this paper. In the supplementary material we present a successful application of our method to a domain with continuous state transitions, using the SAA method described above.

4. Robust Approximate Policy Iteration

In this section we propose a policy improvement algorithm, driven by the RPVI method of the previous section.

First, let us introduce the state-action value function $Q^\pi(x,u) = \inf_{P\in\mathcal{P}} \mathbb{E}^{\pi,P}\left[\sum_{t=0}^\infty \gamma^t r(x_t,u_t) \middle| x_0=x, u_0=u\right]$, which is more convenient for applying the optimization step of policy iteration than $V^\pi(x)$. We assume linear function approximation of the form $\tilde{Q}^\pi(x,u) = \phi(x,u)^\top w$, where $\phi(x,u)\in\mathbb{R}^k$ is a state-action feature vector and $w\in\mathbb{R}^k$ is a parameter vector. Note that $Q^\pi(x,u)$ may be seen as the value function of an equivalent RMDP with states in $\mathcal{X}\times\mathcal{U}$, therefore the policy evaluation algorithm of Section 3 applies. Also, note that given some w, a greedy policy $\pi^*_w(x)$ at state x with respect to that approximation may be computed by

$$\pi^*_w(x) = \arg\max_u \phi(x,u)^\top w, \tag{14}$$

and we write $\phi^*_w(x) = \phi(x,\pi^*_w(x))$, and let Φ^*_w denote a matrix with $\phi^*_w(x)$ in its rows.

The Approximate Robust Policy Iteration (ARPI) algorithm is initialized with an arbitrary parameter vector w_0. At iteration $i+1$, we estimate the parameter w_{i+1} of the *greedy* policy with respect to w_i as follows. We initialize $\theta_0\in\mathbb{R}^k$ to some arbitrary value, and then iterate on θ:

$$\theta_{j+1} = \left(\Phi^\top D\Phi\right)^{-1}\left(\Phi^\top Dr + \gamma\Phi^\top D\sigma_\pi(\Phi^*_{w_i}\theta_j)\right), \tag{15}$$

where the terms in (15) are estimated from data (cf. Eq. 11) according to $\Phi^\top D\Phi \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)\phi(x_t,u_t)^\top$, $\Phi^\top Dr \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)r(x_t,u_t)^\top$, and $\Phi^\top D\sigma_\pi(\Phi^*_{w_i}\theta_j) \sim \frac{1}{N}\sum_{t=0}^{N-1}\phi(x_t,u_t)\sigma_{\mathcal{P}(x_t,u_t)}(\Phi^*_{w_i}\theta_j)$. Note that, similarly to Eq. (11), each term in the last sum requires the solution of the following problem $\inf_{p\in\mathcal{P}(x,u)}\sum_{x\in\mathcal{X}_r(x,u)} p(x)\phi(x,\pi^*_{w_i}(x))^\top\theta_j$, which may be solved efficiently for the uncertainty sets discussed above. After θ has converged, we set w_{i+1} to its final value. In practice, we only iterate (15) for a few iterations[5] and set w_{i+1} to the last value of θ.

[5]Due to the fast convergence of (15) in practice, we didn't employ more sophisticated stopping conditions.

For comparison, in standard LSPI (Lagoudakis & Parr, 2003) the iteration on θ is not needed, as the policy evaluation equation (3) is linear, and may be solved using a least squares approach (LSTD; Boyan 2002). Computationally, the contraction property of Corollary 4 guarantees a linear convergence rate for the θ iteration, therefore the addition of this step should not impact performance significantly. Also, note that the computation of $\Phi^\top D\Phi$ and $\Phi^\top Dr$ only needs to be done once.

For standard approximate policy iteration, a classical result (Bertsekas, 2012) bounds the error (closeness to optimality) of the resulting policy by errors in policy evaluation and policy improvement. We now extend this result to robust approximate policy iteration.

Consider a general approximate robust policy iteration method that generates a sequence of policies $\{\pi_i\}$ and corresponding robust value functions $\{V_i\}$ that satisfy

$$\|V_i - V^{\pi_i}\|_\infty \le \delta, \quad \|T^{\pi_{i+1}}V_i - TV_i\|_\infty \le \epsilon. \tag{16}$$

The following extension of Proposition 2.5.8 of Bertsekas (2012) bounds the error $\|V^{\pi_i} - V^*\|_\infty$. The proof is based on the contraction and monotonicity properties of T^π and T, and detailed in the supplementary material.

Proposition 6. *The sequence $\{\pi_i\}$ generated by the general approximate robust policy iteration algorithm (16) satisfies*

$$\limsup_{i\to\infty}\|V^{\pi_i} - V^*\|_\infty \le \frac{\epsilon + 2\gamma\delta}{(1-\gamma)^2}.$$

Note that in the ARPI algorithm, since we are working with state-action values, and solve the maximization in (14) explicitly, there are no errors in the policy improvement step. We therefore have the following corollary

Corollary 7. *Consider the ARPI algorithm (15), and denote $Q_i(x,u) = \phi(x,u)^\top w_i$ and $\pi_i = \pi^*_{w_{i-1}}$. If the sequence of value functions satisfy $\|Q_i - Q^{\pi_i}\|_\infty \le \delta$ for all i, then $\limsup_{i\to\infty}\|Q^{\pi_i} - Q^*\|_\infty \le \frac{2\gamma\delta}{(1-\gamma)^2}$.*

Corollary 7 suggests that the ARPI algorithm is fundamentally sound. We note that more general L_2-norm bounds for approximate policy iteration were proposed by Munos (2003), and extending them to the robust case requires further work. In addition, Kaufman & Schaefer (2012) provide bounds for robust policy iteration without function approximation, but with errors in the calculation of the $\sigma_{\mathcal{P}(x,u)}$ operator.

5. Applications

In this section we discuss applications of robust ADP. We start with a discussion of optimal stopping problems. Then, we present an empirical evaluation on an option trading domain – a finite horizon continuous state space optimal stopping problem, for which an exact solution is intractable.

An optimal stopping problem is an RMDP where the only choice is when to terminate the process. Formally, the action set is binary $\mathcal{U} = \{0, 1\}$, and executing $u = 1$ from any state always transitions to a terminal state with probability 1 (and no uncertainty). Let $\hat{\pi}$ denote a policy that never chooses to terminate, i.e., $\hat{\pi}(x) = 0, \forall x$. In the supplementary material we show that if Assumption 2 is satisfied for $\pi = \hat{\pi}$, then it is immediately satisfied for all other policies. While this does not ease the conditions that Assumptions 2 places on the uncertainty set and discount factor, it simplifies the design of a suitable exploration policy.

5.1. Option Trading

In this section we apply ARPI to the problem of trading American-style options. An American-style put (call) option (Hull, 2006) is a contract which gives the owner the right, but not the obligation, to sell (buy) an asset at a specified strike price K on or before some maturity time T. Letting the state x_t represent the price of the asset at time $t \leq T$, the immediate payoff of executing a put option at that time is $g_{put}(x_t)$, where $g_{put}(x) \doteq \max(0, K - x)$, whereas for a call option we have $g_{call}(x) \doteq \max(0, x - K)$. Assuming Markov state transitions, an optimal execution policy may be found by solving a finite horizon optimal stopping problem; however, since the state space is typically continuous, an exact solution is infeasible. Even calculating the value of a given policy, an important goal by itself, is challenging. Previous studies (Tsitsiklis & Van Roy, 2001; Li et al., 2009) have proposed RL solutions for these tasks, and shown their utility. Here we extend this approach.

One challenge of option investments is that the underlying model is never truly known, but only accessed through historical data, in the form of state trajectories (e.g., stock prices over time). Catering for risk-averse traders, we plan policies based on the worst-case model that fits the data.

In the following we show that option trading may be formulated as an RMDP, and then present our results of applying the ARPI algorithm to the problem. We consider three different scenarios: a simple put option, a combination of a put and a call, and a case of model misspecification.

5.1.1. An RMDP Formulation

The option pricing problem may be formulated as an RMDP as follows. To account for the finite horizon, we include time explicitly in the state, thus, the state at time t is $\{x_t, t\}$. The action is binary, where 1 stands for executing the option and 0 for continuing to hold it. Once an option is executed, or when $t = T$, a transition to a terminal state takes place. Otherwise, the state transitions to $\{x_{t+1}, t+1\}$ where x_{t+1} is determined by a stochastic kernel $\hat{P}(x'|x, t)$. The reward for executing $u = 1$ at state x is $g(x)$ and zero otherwise. We have $g(x) = g_{put}(x)$ for a put option, $g(x) = g_{call}(x)$ for a call option, or some combination of them for a mixed investment.

Note that the state-action values for execution is known in advance, for we have $Q(\{x, t\}, u = 1) = g(x)$ by definition. Therefore, we only need to estimate the value of not exercising the option. We use linear function approximation $\tilde{Q}^{\pi}(\{x, t\}, u = 0) = \phi(\{x, t\})^{\top} w$, and the ARPI update equation (15) in this case may be written as $\theta_{j+1} = \left(\Phi^{\top} D \Phi\right)^{-1} \left(\gamma \Phi^{\top} D \sigma_{\pi}(\nu)\right)$, where $\nu(x, t)$ equals $g(x)$ if $g(x) > \phi(\{x, t\})^{\top} w_i$, and equals $\phi(\{x, t\})^{\top} \theta_j$ otherwise. As our features we chose 2-dimensional (for x and t) radial basis functions (RBF).[6]

The parameters for the experiments are provided in the supplementary material, and were chosen to balance the different factors in the problem. Most importantly, we chose $\gamma = 0.98$ and a large uncertainty set such that Assumption 2 is *severely violated*. We did not, however, encounter any convergence problems, indicating that our method works well beyond the limits of Assumption 2. The Matlab code for these results is provided in the supplementary material.

5.1.2. Trading with a Put Option

Here we consider a simple put option, where K is equal to the initial price x_0. Our price fluctuation model M follows a Bernoulli distribution[7] (Cox et al., 1979), $x_{t+1} = \begin{cases} f_u x_t, & \text{w.p. } p \\ f_d x_t, & \text{w.p. } 1 - p \end{cases}$, where the up and down factors, f_u and f_d, are constant. Our empirical evaluation proceeds as follows. In each experiment, we generate N_{data} trajectories of length T from the true model M. From these trajectories we form the maximum likelihood estimate of the up probability $\hat{p}$, and the 95% confidence intervals $\hat{p}_-$ and $\hat{p}_+$ using the Clopper-Pearson method (Clopper & Pearson, 1934), which constructs our uncertain model M_{robust}. We also build a model without uncertainty $M_{nominal}$ by setting $\hat{p}_- = \hat{p}_+ = \hat{p}$. Using $\hat{p}$, we then simulate N_{sim} trajectories of length T (this corresponds to a policy that never executes the option), where $x_0 = K + \epsilon$, and ϵ is uniformly distributed in $[-\delta, \delta]$. These trajectories are used as input data for the ARPI algorithm of Section 4.

Let π_{robust} and $\pi_{nominal}$ denote the policies found by ARPI using M_{robust} and $M_{nominal}$, respectively. We evaluate the performance of π_{robust} and $\pi_{nominal}$ using N_{test}

[6] In comparison, Li et al. (2009) used Laguerre polynomials for x and several monotone functions for t. We observed significantly better performance with the RBFs. We attribute this to the non-separable (in x and t) nature of the value function, a property that is not captured by the representation of Li et al. (2009).

[7] Similar results were obtained with a geometric Brownian motion model, using the SAA method for solving the inner problem. These results are provided in the supplementary material.

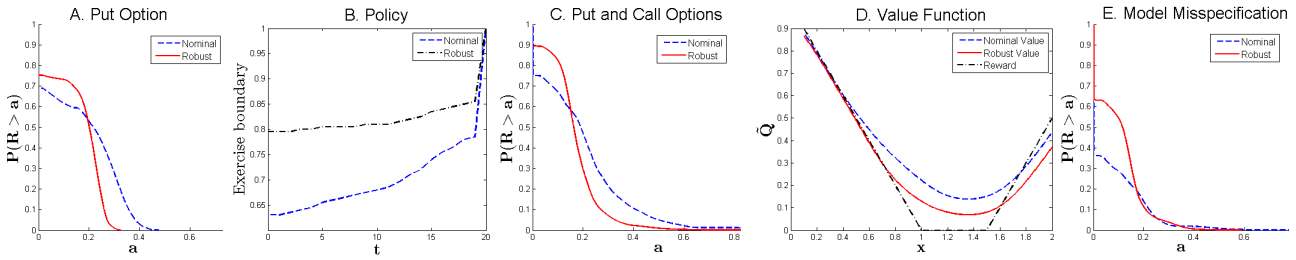

Figure 1. Performance of robust vs. nominal policies. A,C,E: The tail distribution (complementary cumulative distribution function) of the total reward R for the put option (A), put and call (C) and model misspecification (E) scenarios. Note that a higher value for some a indicates a higher chance of guaranteeing a reward of at least a, therefore the plots (A) and (C) display a risk-sensitive behavior of the robust policies. The results were obtained from 100 independent experiments. B: The nominal and robust policies for the put option scenario, represented by the exercise boundary for each t. D: The reward $g(x)$ and value function $\tilde{Q}(x, t = 5)$ from a typical experiment of the put and call option scenario.

trajectories obtained from the *true* model M. Recall that we seek risk-averse policies; thus, the advantage of π_{robust} should reflect in the least favorable outcomes. In Figure 1A we plot the tail distribution of the total reward R (from 100 experiments) obtained by π_{robust} and $\pi_{nominal}$. It may be seen that π_{robust} has a lower probability of obtaining a low payoff (or losing the investment). This, however, comes at a cost of a smaller probability for a high payoff. To the risk-sensitive investor, such results are important. In Figure 1B we further illustrate the policies π_{robust} and $\pi_{nominal}$ by plotting the exercise boundary (the lowest price for which the policy decides to exercise) for each t. The conservative behavior of π_{robust} is evident.

5.1.3. Trading with a Put and a Call

We now consider a more complicated scenario, where the trader has bought both a put option, with strike price $K_{put} < x_0$, and a call option, with strike $K_{call} > x_0$. The reward is given by $g(x) = g_{put}(x) + g_{call}(x)$, and the models and experimental procedure are the same as in the previous scenario. In Figure 1C we plot the tail distribution of the total reward (from 100 independent experiments) obtained by π_{robust} and $\pi_{nominal}$. Notice that the risk-averse policy has a significantly smaller chance of losing the investment. In Figure 1D we display the reward $g(x)$ and the (approximate) value functions $\tilde{Q}^{\pi_{robust}}$ and $\tilde{Q}^{\pi_{nominal}}$ from a typical experiment, for $t = 5$. The robust value function is important by itself, as it holds valuable information about the expected future profit.

5.1.4. Robustness to Model Misspecification

In the previous scenarios we assumed that our estimated models, M_{robust} and $M_{nominal}$, are the same as the true model M. In practice, this is rarely the case, and one has to consider the possibility of model misspecification. An RMDP model provides some robustness against model misspecification, as we now demonstrate. Let the probability p in the *true* model M depend on the state according to $p(x) = p_1 \mathbb{1}\{x \leq \alpha\} + p_2 \mathbb{1}\{x > \alpha\}$, where the threshold α is $(K_{put} + K_{call})/2$. However, let the estimated models M_{robust} and $M_{nominal}$, and the experimental procedure remain as before. We consider again the case of both a put and a call option, as in Section 5.1.3. In Figure 1E we plot the tail distribution of the total reward (from 100 independent experiments) obtained by π_{robust} and $\pi_{nominal}$. Observe that in this case, the misspecification of the nominal model led to a policy that is dominated by the robust policy, which was less affected by this problem.

6. Conclusion and Future Work

We presented a novel framework for solving *large-scale* uncertain Markov decision processes. To the best of our knowledge, such problems are beyond the capabilities of previous studies, which focused on exact solutions and hence suffer from the "curse of dimensionality". We presented both formal guarantees and empirical evidence to the usefulness of our approach. As we demonstrated, uncertain MDPs are suitable for both risk-averseness and mitigation of model misspecification, indicating their importance for decision making under uncertainty.

Interestingly, as was recognized by Iyengar (2005), results on robust MDPs may also be extended to their 'best-case' counterpart, known as optimistic MDPs[8]. Such are useful for efficient exploration, as in the UCRL2 algorithm (Jaksch et al., 2010), suggesting a future extension of our work.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement n. 306638. H. Xu is partially supported by the Ministry of Education of Singapore through AcRF Tier Two grant R-265-000-443-112.

[8] See the supplementary material for more details.

References

Bagnell, A., Ng, A., and Schneider, J. Solving uncertain Markov decision problems. Technical Report CMU-RI-TR-01-25, Carnegie Mellon University, August 2001.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol II*. Athena Scientific, fourth edition, 2012.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Bertsekas, D. P. and Yu, H. Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227(1):2750, 2009.

Boyan, J. A. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Cox, J. C., Ross, S. A., and Rubinstein, M. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.

Gordon, G. J. Stable function approximation in dynamic programming. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

Hull, J. C. *Options, Futures, and Other Derivatives (6th edition)*. Prentice Hall, 2006.

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

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Kaufman, D. L. and Schaefer, A. J. Robust modified policy iteration. *INFORMS Journal on Computing*, 2012.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4: 1107–1149, 2003.

Li, Y., Szepesvari, C., and Schuurmans, D. Learning exercise policies for American options. In *Proc. of the 12th International Conference on Artificial Intelligence and Statistics, JMLR: W&CP*, volume 5, pp. 352–359, 2009.

Mannor, S., Simester, D., Sun, P., and Tsitsiklis, J. N. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Munos, R. Error bounds for approximate policy iteration. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 560–567, 2003.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.

Nilim, A. and El Ghaoui, L. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Powell, W. B. *Approximate Dynamic Programming*. John Wiley and Sons, 2011.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.

Shapiro, A. and Nemirovski, A. On complexity of stochastic programming problems. In *Continuous optimization*, pp. 111–146. Springer, 2005.

Strehl, A. L. and Littman, M. L. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*, pp. 856–863. ACM, 2005.

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

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvari, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

Tsitsiklis, J. N. and Van Roy, B. Regression methods for pricing complex American-style options. *Neural Networks, IEEE Transactions on*, 12(4):694–703, 2001.