
Qualitative Reinforcement Learning

Arkady Epshteyn
Gerald DeJong

AEPSHTEY@UIUC.EDU
DEJONG@UIUC.EDU

Computer Science Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA

Abstract

When the transition probabilities and rewards of a Markov Decision Process are specified exactly, the problem can be solved without any interaction with the environment. When no such specification is available, the agent's only recourse is a long and potentially dangerous exploration. We present a framework which allows the expert to specify imprecise knowledge of transition probabilities in terms of stochastic dominance constraints. Our algorithm can be used to find optimal policies for qualitatively specified problems, or, when no such solution is available, to decrease the required amount of exploration. The algorithm's behavior is demonstrated on simulations of two classic problems: mountain car ascent and cart pole balancing.

1. Introduction

When a Markov Decision Process (MDP) is specified precisely by the domain expert, no exploration of the environment is necessary. Algorithms such as policy iteration and value iteration (Sutton & Barto, 1998) can be used to compute the optimal solution, which may be subsequently applied online. Unfortunately, in many domains it is unrealistic to expect that an expert will be able to come up with precise system dynamics. In such domains, an agent can resort to reinforcement learning (RL) to explore its environment. However, extensive exploration can be undesirable for the following reasons: it is time-consuming, expensive (in terms of wear and tear on the robotic equipment), and perilous when the agent chooses to explore dangerous states (e.g., nuclear reactor meltdown for an agent controlling a nuclear plant or car going off the road for a car-driving agent. Abbeel and Ng also describe a helicopter crash which occurred during overly aggressive exploration (Abbeel & Ng, 2005)).

More importantly, the agent may find itself in new states when interacting with the world which were never encountered during learning (this is an especially common problem in continuous environments). Consider, for example, a car driving agent which drives off the road because it is going too fast while taking a turn. Even if the agent learns that the optimal policy in this situation is to slow down, it will repeat the same mistake when taking a similar turn at an even faster speed. This inability of the agent to transfer acquired information between states does not just increase the amount of exploration required to learn a good policy - it also prevents the agent from acting optimally in parts of the environment unseen during the learning stage (a car driving agent that learns to drive on small hills may have trouble after being transferred to a mountainous terrain, even though the same principles apply).

Notice that, in the above example, simple qualitative statements about the domain of the sort: "a higher mountain is more difficult to climb than a lower mountain", or "a turn is easier to take at a lower speed" may be sufficient to facilitate the kind of reasoning needed to generalize the learning experience and enable the agent to solve the problem without resorting to extensive exploration. However, these statements cannot be expressed in the language of MDP transition probabilities, and can never be fully acquired through reinforcement learning unless one sees every mountain in the world (some of which may be too dangerous to climb).

In this paper, we introduce a framework which allows the expert to specify a set of comparative statements about the domain. This qualitative description of the problem is satisfied by multiple quantitative worlds, with each world describing an MDP with completely specified transition probabilities and rewards. We present an algorithm which, given such a qualitative description, returns a set of policies guaranteed to contain the optimal policy for every possible quantitative instantiation of the description. As an example, we apply our algorithm to the well-known problem of driving a car up a steep mountain (Sutton & Barto,

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

1998), *with the caveat that the output of the car’s engine is corrupted by arbitrary bounded stochastic noise*. Since the optimal policy depends on the engine power, our algorithm can be viewed as a tool for examining the sensitivity of the optimal policy to noise. We also apply the algorithm to another well-studied problem, that of balancing a pole on a cart (Sutton & Barto, 1998), under a similar assumption that the power of the cart’s engine is uncertain.

Our algorithm allows MDP designers to obtain optimal solutions to problems without having to provide completely quantified specifications if the set of policies it returns is small enough to achieve good behavior in most states. If that is not the case, we present a variant of the algorithm which, given a qualitative description of the problem, combines it with limited exploration to discover the optimal policy much faster than traditional reinforcement learning, and the policy that it discovers is more broadly applicable.

The rest of the paper is organized as follows: we describe related work in Section 2. In Section 3, we describe the variant of MDPs which we study. In Sections 4 and 5, our framework of qualitative MDPs (QMDP) and qualitative reinforcement learning (QRL) is described. Experiments are presented in Section 6, followed by conclusions in Section 7.

2. Related Work

Qualitative Markov Decision Processes have been studied by Bonet and Pearl (Bonet & Pearl, 2002) and Sabbadin (Sabbadin, 1999). Bonet’s study is purely theoretical, Sabbadin describes an application of his algorithm to a 3×3 gridworld. By contrast, we describe experiments with our algorithm on much more realistic problems which are an order of magnitude bigger than the 3×3 gridworld. More importantly, there is no clear connection between qualitative representations of MDPs proposed in these two papers and quantitative probabilities which can be estimated via empirical interaction with the environment. For this reason, neither study attempts to combine the qualitative problem description with quantitative exploration. In our approach, qualitative statements have a clear probabilistic interpretation, which enables us to construct such a combination.

Several ways of limiting exploration in reinforcement learning with prior knowledge have been proposed. Shaping (Ng et al., 1999; Laud & DeJong, 2003) attempts to direct the agent to explore regions which are likely to lead to good solutions by modifying the reward function. However, it may be difficult to deter-

mine a-priori which states will ultimately lead to good solutions and which should not be explored. Apprenticeship learning (Abbeel & Ng, 2005; Abbeel & Ng, 2004) is a framework in which an agent learns the expert’s reward function by observing his demonstrated behavior, thus avoiding direct interaction with the environment. The advantage of our approach is that our model of prior knowledge only requires specifying how the world works, not how to explore it. It may be used in domains where the expert has pertinent information about the world dynamics, but does not know how to solve the problem.

An alternative approach to dealing with uncertainty in the specification of MDPs without resorting to exploration is the minimax robustness framework (see e.g., (Givan et al., 2000)). In this framework, the agent is also presented with a description of the world which corresponds to a set of completely specified MDPs. The agent’s goal is to select the best optimal policy, knowing that for any policy the agent selects, adversarial nature will choose the worst possible world in which to evaluate it. In our framework, on the other hand, the agent seeks a set of policies which contains the optimal one for every possible completely specified MDP.

3. Preliminaries

Instead of regular Markov Decision Processes, in what follows we use a variant of MDPs in which the agent is only interested in the nearest reward. A reward received later is foreseen for any probability of receiving any reward earlier, and bigger expected rewards are preferred to smaller rewards received at the same time. Ties between rewards to be received after n steps are broken by looking at rewards to be received after $n+1$ steps, once again preferring bigger rewards to smaller, and so on, up to N steps ahead. We will refer to this MDP as Myopic MDP because of its strong preference for receiving rewards sooner. In Section 6, we present experimental evidence that this variant gives reasonable policies for control problems.

In order to formalize this paradigm, we use the framework of generalized Markov Decision Processes. A generalized finite Markov Decision Process is a tuple $(S, A, P, R, \otimes, \oplus, Next)$, where S is a finite set of states, A is a finite set of actions, R is a reward function, $P : S' | S \times A \rightarrow [0, 1]$ is a transition probability function, $Next : S \times A \rightarrow S$ is a set of states reachable with nonzero probability in one step after taking action $a \in A$ in state $s \in S$, a summary operator \oplus defines the value of transitions based on the value of the successor states, and a summary operator \otimes de-

fines the value of a state based on the values of all state-action pairs. These operators are used to define the generalized form of Bellman's equation as follows: for each state s , the optimal value function $V^*(s) = [H[KV^*]](s)$, where $[KV](s, a) = R(s, a) + \oplus_{s'}^{(s,a)} V(s')$ and $[HV](s) = \otimes_a^{(s)}([KV](s, a))$. Setting $\oplus_{s'}^{(s,a)} g(s') = \alpha \sum_{s'} P(s'|s, a)g(s')$ and $\otimes_a^{(s)} f(s, a) = \max_a f(s, a)$ recovers the conventional MDP formulation with the discount factor $\alpha \in (0, 1)$. In our myopic framework, on the other hand, the value function $V_\pi(s)$ for a fixed policy $\pi : S \rightarrow A$ is a N -dimensional vector, with each component $V_{\pi,i}, i \in 1, \dots, N$ representing the expected positive reward the agent will receive i steps after starting out in state s and following π^1 . Similarly, the reward $R(s, a) = [r(s, a), 0, 0, \dots, 0]$ is a reward vector indicating that reward $r(s, a) \geq 0$ is received for choosing action a in state s . The summary operators are defined to facilitate correct propagation of rewards: $\oplus_{s'}^{(s,a)} g(s') = \sum_{s'} P(s'|s, a)[0, g(s')]$ propagates rewards back from the successor states. Before we define the \otimes operator, we need to impose an order relation on the values of states. This is done with the following definition:

Definition 3.1. Let $U \in \mathbb{R}^n$ and $V \in \mathbb{R}^n$ be two componentwise non-negative n -dimensional vectors.

$$\text{Let } f^*(U, V) = \min_{i: U_i > V_i \geq 0 \text{ or } V_i > U_i \geq 0} i$$

be the smallest component, the value of which is strictly greater in one of the vectors than in the other one. Then define $U \prec V \Leftrightarrow \{(f^* \text{ exists}) \wedge (U_{f^*} < V_{f^*})\}$, $U = V \Leftrightarrow f^*$ does not exist, and $U \preceq V \Leftrightarrow \{U \prec V \vee U = V\}$.

This order instantiates the myopic comparison of values of two actions. If $U(s)$ and $V(s)$ represent the values of two policies executed in s , then $f^*(U(s), V(s))$ is the first time step in which expected rewards of following U and V differ, and \prec prefers the policy with larger reward in this time step. It can be verified that \preceq is a total order, which means that a maximum is well-defined for any finite set of vectors. We take the \otimes operator to be this maximum: $\otimes_a^{(s)} f(s, a) = \max_a f(s, a)$.

The optimal value function $V^*(s)$ which satisfies Bellman's equation can be computed policy iteration, which consists of policy evaluation (which computes $V_\pi^{t+1}(s) = [KV_\pi^t](s, \pi(s))$ for a fixed policy π) interleaved with policy improvement, which updates $V^{t+1}(s) = [HV_\pi^t](s)$. The following theorem shows

¹ N is the horizon of the MDP. All of our results assume that N is large enough and still hold as $N \rightarrow \infty$.

that for the Myopic MDP, policy evaluation and policy iteration converge:

Theorem 3.2. For the Myopic MDP, there is a unique optimal value function V^* which satisfies the myopic Bellman's equation. Policy iteration converges to V^* . Moreover, for any fixed policy π , there is a unique optimal value function V_π^* which satisfies $V_\pi^* = KV_\pi^*$, and policy evaluation converges to V_π^* .

Proof. (sketch) The proof is based on showing equivalence between the Myopic MDP policy iteration and policy iteration for a conventional MDP with sufficiently low discount factor α . See the full paper (Epshteyn & DeJong, 2006) for complete proofs of theorems in this paper. \square

In order to model qualitative knowledge, we rely on the notion of first-order stochastic dominance (Shaked & Shanthikumar, 1994), defined as:

Definition 3.3. Let $G = \{g_1, \dots, g_n\}$ be a support set for probability distributions P_1 and P_2 . Let O be a partial order relation on G and define $\overline{O}(y) = \{x \in G : yOx\}$ to be the set of elements of G at least as good as y according to O (similarly, $\underline{O}(y) = \{x \in G : xOy\}$ is the set of elements no better than y according to O).

We say that P_1 stochastically dominates P_2 with respect to O if $\forall y \in G, P_1(\overline{O}(y)) \geq P_2(\overline{O}(y))$ where $P(S) = \sum_{s \in S} P(s)$ is the probability of a set. If, in addition, $\exists z \in G : P_1(\overline{O}(z)) > P_2(\overline{O}(z))$, we say that P_1 strictly stochastically dominates P_2 with respect to O (e.g., using \leq for O gives first-order stochastic dominance on the real line).

4. Qualitative MDP

The policy evaluation step of the policy iteration algorithm for Myopic MDPs has the following useful monotonicity property:

Lemma 4.1. Suppose $V^0 = \mathbf{0}$. Let s_1 and s_2 be any two states. If $V^t(s_1) \prec V^t(s_2)$, then for all subsequent iterations $q > t$ of policy evaluation, $V^q(s_1) \prec V^q(s_2)$.

Proof. (sketch): By induction on t , for any s and for any $j \leq t$, $V_j^t(s)$ (the component of $V^t(s)$ which represents the discounted expected reward received after j steps) does not change after step t . Moreover, $V_j^t(s) = 0$ for any $j > t$. If $V^t(s_1) \prec V^t(s_2)$, then by Definition 3.1 $\exists f^* : (V_f^t(s_1) = V_f^t(s_2), \forall f < f^*) \wedge (V_{f^*}^t(s_1) < V_{f^*}^t(s_2))$. Since $V_j^t(s_1) = 0$ for any $j > t$, $f^* \leq t$. \square

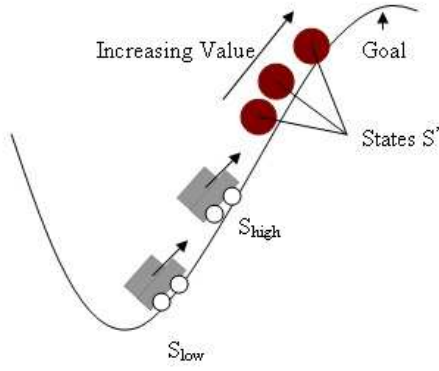


Figure 1. Mountain Car Domain

Lemma 4.1 suggests a qualitative policy iteration algorithm which only keeps track of the ordering of values of states, but not the values themselves, and, similarly, only requires the ordering of rewards as an input instead of the actual values. The pseudocode for the algorithm is given in Figure 2. The key distinction between this algorithm and conventional policy iteration is that it works with pairs of states rather than single states. In each iteration, it updates the ordering between every pair of states based on new ordering information received from the previous iteration. In doing so, it relies on the domain oracle which, given an ordering of states $s' \in \cup_{i=1}^2 \text{Next}(s_i, a_i)$, returns ' $<$ ' if $P(s'|s_1, a_1)$ strictly stochastically dominates $P(s'|s_2, a_2)$ with respect to the given ordering and ' $>$ ' if $P(s'|s_2, a_2)$ strictly stochastically dominates $P(s'|s_1, a_1)$. The domain oracle can also indicate that neither [state, action] pair stochastically dominates the other one (or that it lacks knowledge to indicate dominance) by returning the unknown indicator '?. If $P(s'|s_2, a_2)$ and $P(s'|s_1, a_1)$ stochastically dominate each other, the oracle returns '='. Similarly, the reward oracle returns ' $<$ ($>$, $=$)' if $r(s_1, a_1) <$ ($>$, $=$) $r(s_2, a_2)$. While a domain oracle may seem hard to construct, we will demonstrate such oracles for two realistic control problems in Section 6. An example of a domain oracle in action is shown in Figure 1. It shows a car ascending a mountain, in two positions, one higher and one lower, moving with the same velocity. A possible ordering of next states appears as well, with states higher up the mountain being more valuable. With respect to this ordering, the car in the lower position on the mountain s_{low} has less of a chance of reaching more valuable states than the car in s_{high} and, therefore, $P_{\pi}(s'|s_{high})$ stochastically dominates $P_{\pi}(s'|s_{low})$ for any policy π .

SAMEORDER(ORACLE, $Order$, s_1 , s_2 , Π_1 , Π_2)
Input: Procedure ORACLE, $Order$ on S , States $s_1 \in S$, $s_2 \in S$, Sets of actions Π_1, Π_2
Output: $order \in \{ '<', '>', '=', '?'\}$

1. **if** ORACLE($Order$, s_1 , a_1 , s_2 , a_2) returns the same value $order$ for all pairs of actions $a_1, a_2 \in \Pi_1 \times \Pi_2$
2. **then return** $order$
3. **else return** '?'

POLICY EVALUATION(Set of policies Π)

1. $j \leftarrow 0$
2. **for all** pairs of states $s_1, s_2 \in S \times S$
3. **do** $Step_Order^j \leftarrow$ SAMEORDER (REWARD_ORACLE, \emptyset , $s_1, s_2, \Pi(s_1), \Pi(s_2)$)
4. **repeat**
5. **for all** pairs of states $s_1, s_2 \in S \times S$
6. **do** $order \leftarrow$ SAMEORDER (ORACLE, $Step_Order^j, s_1, s_2, \Pi(s_1), \Pi(s_2)$)
7. $Step_Order^{j+1}(s_1, s_2) \leftarrow order$
8. **if** $Order(s_1, s_2) \neq order$
9. **then** $Order(s_1, s_2) \leftarrow order$
10. $j \leftarrow j + 1$
11. **until** $Order$ stops changing
12. **return** $Order$

POLICY IMPROVEMENT($Order$ on S)

1. **for all** states $s \in S$
2. **do** $best_actions \leftarrow \emptyset$
3. **for all** actions $a, a' \in A_{x \text{ best_actions}}$
4. **do if** ORACLE($Order$, s , a , s , a') $\neq '>'$
5. **then** $best_actions \leftarrow \{best_actions \cup \{a\}\} \setminus \{a'\}$
6. **if** ORACLE($Order$, s , a , s , a') $\neq '?'$
7. **then** $best_actions \leftarrow best_actions \cup \{a\}$
8. $\Pi(s) \leftarrow \emptyset$
9. **for all** actions $a \in best_actions$
10. **do** $\Pi(s) \leftarrow \Pi(s) \cup \{a\}$
11. **return** Π

POLICY ITERATION()

1. Select arbitrary initial policy π
2. \forall states $s \in S : \Pi(s) \leftarrow \{\pi(s)\}$
3. **repeat**
4. $Order \leftarrow$ POLICY EVALUATION(Π)
5. $\Pi \leftarrow$ POLICY IMPROVEMENT($Order$)
6. **until** Π stops changing

Figure 2. Qualitative Policy Iteration Algorithm

Qualitative policy iteration is analogous to conventional policy iteration, with $Order$ replacing the value function, and a set of deterministic candidate policies Π playing the role of the optimal policy. $\Pi : S \rightarrow 2^A$

is represented as a mapping from states to sets of actions, with each action $a \in \Pi(s)$ being possibly optimal in some quantitative instantiation of the qualitative MDP. The following theorem states that, when the qualitative policy iteration algorithm terminates, the optimal policy for any quantitative MDP consistent with the qualitative domain theory is contained in the returned candidate set of policies Π :

Theorem 4.2. *If qualitative policy iteration is executed in parallel with myopic policy iteration on any quantitative MDP consistent with the domain theory, at the end of iteration t , the candidate policy set Π contains the policy returned by the conventional policy iteration algorithm at the end of t .*

Proof. (sketch) For a fixed policy π , the ordering of values Order at iteration i of qualitative policy evaluation corresponds to the ordering of values of Myopic policy iteration at i (according to \prec). This can be seen by induction, with the base case given by the ordering of rewards, and the inductive step implied by Lemma 4.1 and the fact that, for any nonnegative monotonically increasing function $V(s')$ with respect to order O on s' , $\sum_{s'} P_1(s')V(s') < \sum_{s'} P_2(s')V(s')$ if P_2 strictly stochastically dominates P_1 with respect to O . This last fact also implies that qualitative policy improvement does not eliminate the policy chosen by conventional policy iteration. \square

Thus, the set of candidate policies returned by the algorithm on termination is guaranteed to contain the optimal policy.

5. Qualitative RL

When the set of returned policies is too large to be useful for the states of interest, an alternative is to combine qualitative specification of the problem with quantitative probability estimation. This is possible because stochastic dominance constraints have a precise probabilistic interpretation which can be useful for transferring knowledge between states via estimated probabilities. Consider the mountain car example in Figure 1. A priori, we may have reason to believe that all the states in S' are reachable from either s_{low} or s_{high} because the uncertainty in the power range is large enough to allow all of these transitions. Suppose however, that the agent discovers through experimentation that the highest state in S' is not reachable from s_{high} . Since we know that $P(s'|s_{high})$ stochastically dominates $P(s'|s_{low})$, it follows directly from the definition of stochastic dominance that it is not possible to reach that same highest state from s_{low} . Thus, the probability distribution of this, previously

```

PROPAGATE( $order, s_1, a_1, s_2, a_2$ )
Input:  $order \in \{ '<', '>', '=, '? \}$ , States  $s_1 \in S, s_2 \in S$ ,
        Actions  $a_1 \in A, a_2 \in A$ 
1. switch  $order$ 
2.   case  $'<'$ :
3.      $Y \leftarrow \{y \in S : \widehat{P}(\overline{\text{Order}}(y)|s_2, a_2) = 0\}$ 
4.      $i \leftarrow 1$ 
5.   case  $'>'$ :
6.      $Y \leftarrow \{y \in S : \widehat{P}(\underline{\text{Order}}(y)|s_1, a_1) = 0\}$ 
7.      $i \leftarrow 2$ 
8.   default :  $Y \leftarrow \emptyset$ 
9.   for all  $y \in Y$ 
10.    do  $\widehat{P}(y|s_i, a_i) \leftarrow 0$ 
11.     $\text{Next}(s_i, a_i) \leftarrow \text{Next}(s_i, a_i) \setminus \{y\}$ 
    
```

```

QUAL ESTIMATION( $Observed, Order$ )
Input: Set of  $Observed$  transitions  $[s, a]$ ,  $Order$  on  $S$ 
1. for all pairs of states  $s_1, s_2 \in S \times S$ 
2.   do for all pairs of actions  $a_1, a_2 \in \Pi(s_1) \times \Pi(s_2)$ 
3.     do  $order \leftarrow \text{ORACLE}(Order, s_1, a_1, s_2, a_2)$ 
4.     if  $[s_2, a_2] \in Observed \wedge [s_1, a_1] \notin Observed$ 
5.       then PROPAGATE( $order, s_1, a_1, s_2, a_2$ )
    
```

Figure 3. Qualitative Estimation Procedure

unseen transition can be updated with this new piece of knowledge. The following proposition summarizes the inferences which can be made about unknown transition probabilities based on stochastic dominance constraints:

Proposition 5.1. *Let $P_1(s)$ stochastically dominate $P_2(s)$ with respect to some order O . Then if, for some y , $P_1(\overline{O}(y)) = 0$, then $P_2(y) = 0$. If, for some y , $P_2(\underline{O}(y)) = 0$, then $P_1(y) = 0$.*

Proof. The first statement follows immediately from Definition 3.3. The second statement follows from the first and the fact that if $P_1(s)$ stochastically dominates $P_2(s)$ with respect to O , then reversing the order O results in $P_2(s)$ stochastically dominating $P_1(s)$ with respect to reversed O . \square

This observation leads to the following estimation procedure: suppose that the agent has acquired estimates $\widehat{P}(s'|s, a)$ of probabilities of some transitions $[s, a]$ through interaction with the environment. Then the estimation algorithm presented in Figure 3 performs probability estimation based on Proposition 5.1. This algorithm is interleaved with the steps of qualitative policy evaluation (shown in Figure 2).

6. Experiments

Two well-known domains were used to test the qualitative MDP algorithm: mountain car ascent and cart pole balancing. In the mountain-car task, the problem is to drive a car up a steep mountain (see Figure 1). The optimal policy depends on the power of the car’s engine. If the engine is powerful enough to overcome gravity and drive the car up the slope to its goal, the optimal policy is to move towards the goal. Otherwise, the optimal policy is to move away from the goal up the opposite slope, and then apply full throttle to move towards the goal with the help of built-up inertia². The agent received a reward of 1 upon reaching the goal. The task in the cart-pole problem is to balance a pole on a moving cart³. The reward of winding up in state s' was set to $\cos\theta_t(s')$ to encourage actions which keep the pole as upright as possible. In both problems, actions which moved the agent out of bounds of the state space were disallowed.

The mountain car problem exhibits delayed rewards, while in the cart-pole problem rewards are immediate. Both of these problems represent a physical system which keeps track of its continuous state $z^t = [z_1^t, \dots, z_n^t]$ at time t through the update equations $z^{t+1} = [T_1(F; z^t), \dots, T_k(F; z^t), T_{k+1}(z^t), \dots, T_n(z^t)]$ which define the system’s behavior under the influence of the input force F . In the simulation, the state space is discretized, with the boundary of the grid cell for each discrete state s given by $[s_i, \bar{s}_i], i \in \{1, \dots, n\}$. The dynamics of the system are simulated by picking a characteristic position $z = Z(s)$ in each grid cell s and simulating each action from this position.

²The dynamics of car motion in terms of its position x_t and velocity \dot{x}_t are given by the following equations:

$x_{t+1} = x_t + \dot{x}_{t+1}$, $\dot{x}_{t+1} = \dot{x}_t + Fa_t - G\cos(3x_t)$, where F is the amount of force applied by the engine, G is the pull of gravity, and $a_t \in \{+1$ (full throttle forward), -1 (full throttle reverse), and 0 (zero throttle)} is the action. We used $G = 1$ in our experiments. The state space was discretized by an 11×21 grid in the bounds $-1.2 \leq x \leq 0.5$, $-0.07 \leq \dot{x} \leq 0.07$.

³The state in the cart-pole problem is described by the angle between the pole and the vertical θ_t , the velocity of the cart \dot{h}_t , and the velocity of the pole $\dot{\theta}_t$. The update equations are:

$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t [-Fa_t - m_p l \dot{\theta}_t^2 \sin\theta_t] / (m_c + m_p)}{l[4/3 - m_p \cos^2(\theta_t) / (m_c + m_p)]}$, $\ddot{h}_t = \frac{Fa_t + m_p l [\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t]}{m_c + m_p}$, $\dot{h}_{t+1} = \dot{h}_t + \tau \ddot{h}_t$, $\theta_{t+1} = \theta_t + \tau \dot{\theta}_t$, $\dot{\theta}_{t+1} = \dot{\theta}_t + \tau \ddot{\theta}_t$, with gravity $g = 9.8$, cart mass $m_c = 1$, pole mass $m_p = 0.1$, distance from center of mass of the pole to the pivot $l = 0.5$, time step $\tau = 0.02$, and F is the force, $a_t \in -1, 1$ is the action. The state space was discretized into an $8 \times 8 \times 8$ grid in the range $-1.15 \leq \dot{h} \leq 1.15$, $-0.21 \leq \theta \leq 0.21$, $-2 \leq \dot{\theta} \leq 2$.

Qualitative MDP can be used to capture the situation when the engine’s power is corrupted by unknown, but bounded noise. The power is modeled by a stochastic variable, distributed according to some unknown probability density function (pdf) with known support interval $[I_1, I_2]$ on which the pdf is strictly positive. We will show how to construct the domain oracle automatically under the assumption that the dynamics of the system are specified by invertible functions of the input force F , $T_i(F; z^t), i = 1, \dots, k$ and constant functions of F $T_i(z^t), i = k+1, \dots, n$. Both the mountain car and the cart-pole dynamics can be expressed in terms of such functions (see (Epshteyn & DeJong, 2006)).

For any $[s, a, s']$ tuple, we can determine the range of forces $\Phi(s'|s, a)$ under which applying action a in state s transitions the system to s' by inverting the transition dynamics as follows: $\Phi(s'|s, a) = \bigcap_{i=1}^k [T_i^{-1}(s'_i; Z(s)), T_i^{-1}(\bar{s}'_i; Z(s))] \cap [I_1, I_2]$ if $\forall i \in \{k+1, \dots, n\}, T_i(Z(s)) \in [s'_i, \bar{s}'_i]$ (assuming monotonically increasing $T_i(F; z^t)$, bounds are reversed for monotonically decreasing $T_i(F; z^t)$).

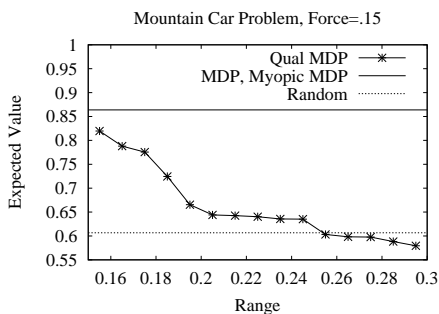
In order to handle sets of states S' , we define $\Phi(S'|s, a) = \bigcup_{s' \in S'} \Phi(s'|s, a)$ as the set of forces under which applying action a in s transitions the system to one of the states $s' \in S'$. The next proposition states that stochastic dominance of probability distributions can be determined by checking the subset relationship for ranges of forces:

Proposition 6.1. *Let $\bar{O}(y) \cap Next(s, a)$ denote the set of next states for the transition $[s, a]$ which are at least as good as y with respect to order O . If $\forall y \in S, \Phi(\bar{O}(y) \cap Next(s_1, a_1)|s_1, a_1) \subseteq \Phi(\bar{O}(y) \cap Next(s_2, a_2)|s_2, a_2)$, then $P(s'|s_2, a_2)$ stochastically dominates $P(s'|s_1, a_1)$. Strict stochastic dominance holds when the subset is proper for some y .*

Proof. By property of probability, $A \subseteq B \Rightarrow P(A) \leq P(B)$, and $A \subset B \subseteq [I_1, I_2] \Rightarrow P(A) < P(B)$ by strict positivity of $P(x)$ on its support interval. \square

In the first set of experiments, qualitative policy iteration was applied to a mountain car problem with the set of possible next state transitions constructed based on the power support interval $[0.15 - r, 0.15 + r]$ around the force $F = 0.15$. This power is insufficient to overcome the force of gravity, so the optimal policy has to move the car up the opposite slope first. Results are presented in Figure 4-(a), which compares the values of the random policy, the optimal policy for $F = 0.15$ for the conventional discounted MDP (with the discount factor $\alpha = 0.9$), the optimal policy for the same F for the Myopic MDP, and the set of policies

a)



b)

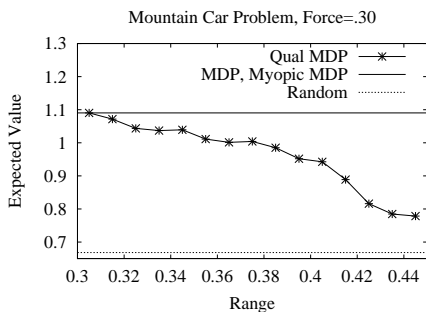


Figure 4. Performance of Qualitative Policies for Mountain-Car Task. Plot of expected value of the policy versus the upper bound I_2 of the power support.

Π computed via qualitative policy iteration. The expected value of each evaluated policy π was measured by the expected discounted return (with discount factor $\alpha = 0.9$) an agent would receive by randomly selecting a starting state s and following π thereafter. Set of policies Π was evaluated by choosing actions uniformly at random from $\Pi(s)$ in each state s . Qualitative policy iteration was evaluated on the increasing range r of the support interval. The policy degrades with increasingly noisy power as Π becomes very large. For a wide range of power support intervals, the qualitative policy performs much better than random and, as the uncertainty interval decreases, its performance approaches that of the optimal MDP policy. Notice that for large power support sets, the qualitative policy (which is correct in some states and random in others) can be outperformed by the completely random policy. A similar experiment was repeated with $F = 0.3$, with similar results shown in Figure 4-(b). This force is large enough to overcome gravity and move the car directly to the goal from the bottom of the valley. The results of qualitative policy iteration for the pole-balancing task based on the increasing uncertainty range around $F = 35$ shown in Figure 5 are similar to the mountain car experiment. In the mountain car problem, the Myopic MDP performed as well

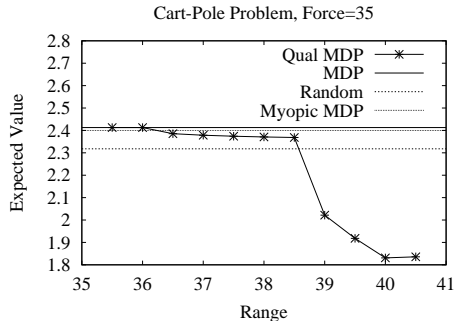


Figure 5. Performance of Qualitative Policies for Pole-Balancing Task. Plot of averaged expected value of the policy versus the upper bound I_2 of the power support.

as conventional MDP, and in the cart-pole problem, the performance gap (due to a more complex reward structure) was negligible.

An experiment was also performed to determine sensitivity of the optimal policy to noise. The support interval for F was set to $[I_1, 0.35]$, with the lower bound I_1 starting at 0.33 and gradually decreasing to observe the degradation in certainty in the optimal policy in different states. The plot of the highest value I_1 at which the set of candidate policies returned by qualitative policy iteration contained more than one action for that state is shown in Figure 6. The optimal policy for the states in the plateau regions never becomes uncertain because in those states, only one action is valid. A more interesting effect is the decreasing ridge of the uncertainty function - it shows that, as the amount of noise in F increases, the policy in states closest to the goal (but with car moving away from the goal) become uncertain first. States farther away from the goal (i.e., on the opposite slope) have enough potential energy to reach the goal with the forward throttle, even if the power is low, so the policy in those states stays certain longer. Thus, qualitative policy iteration can be used to determine robustness of the optimal policy to noise in different parts of the state space.

Finally, we experimented with qualitative reinforcement learning on the mountain car problem. A-priori, the engine power was specified with the uncertainty interval $[0.15, 0.3]$ - too large to determine whether moving towards the goal or away from it is optimal on the bottom of the valley. The actual simulation applied forces to the throttle chosen uniformly from the interval $[0.15, 0.16]$ in each state (for which moving away from the goal was optimal). Each experimental episode started with the car at the bottom of the valley and terminated when it reached the goal state or a state with no valid actions. Whenever $P(s'|s, a)$ was

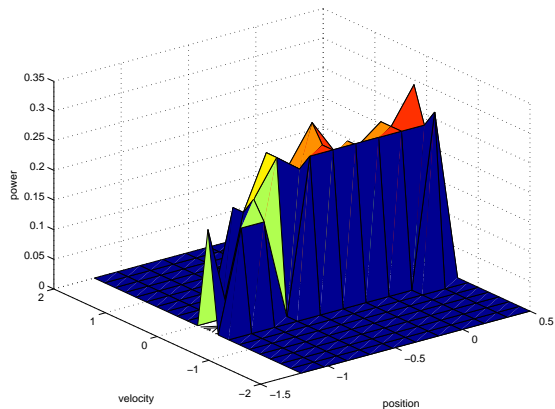


Figure 6. Sensitivity of Qualitative Policy as a function of states.

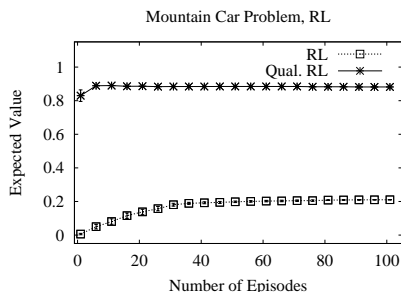


Figure 7. Performance of Qualitative and Conventional RL on the mountain car task. The performance is displayed with 95% error bars based on 100 different episodes for conventional reinforcement learning and 10 different episodes for qualitative reinforcement learning.

estimated to be zero (either directly or through Proposition 5.1), s' was removed from the set $Next(s, a)$, prompting the domain oracle to disregard the forces $\Phi(s'|s, a)$ which could result in a transition to s' when action a was executed in s . Thus, QRL's generalization ability is due to the domain oracle performing a form of estimation of the power interval. Performance of QRL was compared with that of conventional model-based RL which applied policy iteration to estimated transition probabilities (actions in unencountered states were picked randomly). Results are shown in Figure 7 as a function of the number of training episodes. Note that the episodes started out in the same state, but the performance metric averages over the random choice of an initial state. This metric reflects the algorithm's ability to generalize to unseen states. Since some states were not reachable from the starting state due to discretization of time and space, neither algorithm saw them, but QRL deduced how to act in them by comparing them with encountered states.

7. Conclusion

In many MDP problems, it is desirable to avoid exploration as an expensive and potentially dangerous process. We presented an algorithm which either completely eliminates the need to explore while requiring a much less precise description of the problem than an MDP, or limits the amount of exploration needed to act optimally.

Acknowledgements. This work was supported by the Information Processing Technology Office of DARPA under award HR0011-05-1-0040 and in part by the NSF under Award NSF IIS 04-13161. Any opinions, findings, and conclusions expressed in this publication are those of the authors and do not necessarily reflect the views of the DARPA or the NSF.

References

- Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *ICML*.
- Abbeel, P., & Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. *ICML*.
- Bonet, B., & Pearl, J. (2002). Qualitative mdps and pomdps: An order-of-magnitude approximation. *UAI*.
- Epshteyn, A., & DeJong, G. (2006). Qualitative reinforcement learning (full paper). http://www.ews.uiuc.edu/~aepshtey/pubs/qual_rl.ps.
- Givan, R., Leach, S., & Dean, T. (2000). Bounded-parameter markov decision processes. *Artificial Intelligence*, 122, 71–109.
- Laud, A., & DeJong, G. (2003). The influence of reward on the speed of reinforcement learning: An analysis of shaping. *ICML*.
- Ng, A. Y., Harada, D., & Russell, S. J. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*.
- Sabbadin, R. (1999). A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. *UAI*.
- Shaked, M., & Shanthikumar, J. G. (1994). *Stochastic orders and their applications*. San Diego, CA: Academic Press.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.