Learning Without State-Estimation in Partially Observable Markovian Decision Processes

Satinder P. Singh
singh@psyche.mit.edu

Tommi Jaakkola
tommi@psyche.mit.edu

Michael I. Jordan
jordan@psyche.mit.edu

Department of Brain and Cognitive Sciences (E10)
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

Reinforcement learning (RL) algorithms provide a sound theoretical basis for building learning control architectures for embedded agents. Unfortunately all of the theory and much of the practice (see Barto *et al.*, 1983, for an exception) of RL is limited to Markovian decision processes (MDPs). Many real-world decision tasks, however, are inherently non-Markovian, i.e., the state of the environment is only incompletely known to the learning agent. In this paper we consider only partially observable MDPs (POMDPs), a useful class of non-Markovian decision processes. Most previous approaches to such problems have combined computationally expensive state-estimation techniques with learning control. This paper investigates learning in POMDPs without resorting to any form of state estimation. We present results about what TD(0) and Q-learning will do when applied to POMDPs. It is shown that the conventional *discounted* RL framework is inadequate to deal with POMDPs. Finally we develop a new framework for learning without state-estimation in POMDPs by including stochastic policies in the search space, and by defining the value or utility of a *distribution* over states.

1 INTRODUCTION

A diverse variety of sequential tasks of interest to artificial intelligence researchers can be formulated abstractly as embedded agents seeking to control their environment by executing actions. The agents are usually equipped with sensors that provide information about the state of the environment. Reinforcement learning (RL) techniques provide a sound theoretical basis for building learning control architectures for embedded agents (Barto *et al.*, to appear, 1990). Unfortunately all of the elegant theory of RL is limited to Markovian decision processes (MDPs) (Sutton, 1988; Watkins and Dayan, 1992; Dayan, 1992, Jaakkola *et al.*, 1994, Tsitsiklis, to appear). Formulating a given problem as an MDP requires that the agent's sensors return the complete state of the environment. The word state is used here as in control theory to mean all the information necessary to make the prediction of the future states of the environment dependent only on the current state and the future actions and independent of the past states.

While there are interesting problems that can be formulated as MDPs, a great many real-world decision problems have *hidden state*, i.e., are inherently non-Markovian. One can always apply RL algorithms developed specifically for Markovian processes to non-Markovian decision processes (N-MDPs) simply by treating the agent's sensor readings as state descriptions. There is some empirical evidence that such a technique can work well on particular non-Markovian problems (e.g., Barto *et al.*, 1983). However, as yet there is no theory of RL for N-MDPs, and no characterization of the class of N-MDPs on which conventional RL algorithms will perform reasonably well. The general hope that the performance of RL algorithms will degrade gracefully as the degree of non-Markovianness is increased in a given decision problem is unfounded, because it is easy to construct decision problems, where failure to distinguish between just two states can lead to an arbitrarily high absolute loss in performance (see Section 3.1 for an example; also, see Whitehead, 1992).

We show why it is difficult to extend the conventional *discounted* RL framework to environments with hidden state. We present results about what TD(0) (Sutton, 1988) and Q-learning (Watkins, 1989) will do when applied to a class of N-MDPs. Finally we develop a new framework for learning without state-estimation in such N-MDPs by including stochastic policies in the search space, and by defining the value, or utility, of a distribution over states.

2 PREVIOUS APPROACHES

Previous approaches to learning in N-MDPs have focused on methods that combine some form of state-estimation with learning control. Such approaches build an internal representation of the state of the environment by combining sensor readings with past internal representations (Whitehead and Lin, 1993). Several different forms of internal representations have been used: tapped-delay line representations for higher order Markov problems (e.g., Lin and Mitchell, 1992), recurrent neural network based representations (Lin and Mitchell, 1992), and probability distributions over an underlying state space based on the theory of partially observable MDPs (Sondik, 1978; Chrisman, 1992a, 1992b; McCallum, 1993). In addition, Whitehead and Ballard (1990) have proposed using perceptual actions in robots to gather multiple sensor readings, one of which is selected as representing the state of the environment.

A common drawback of all the above methods is that the state estimation component is always based on strong assumptions about the environment. For example, it is usually assumed that the number of states is known in advance. A further drawback is that state estimation is computationally expensive and can require a large amount of data. Even if the true environment has a finite number of states, using state-estimation can result in a continuous space of estimated states making the search problem difficult (e.g., Sondik, 1978). Also the computations performed by the learning control component are wasted until the state-estimation component becomes accurate enough to be useful. Finally, in learning policies that map estimated states to actions, such methods depart fundamentally from conventional RL architectures that learn *memory-less* policies, i.e., learn policies that map the immediate observation of the agent into actions. This paper studies memory-less policies in a class of N-MDPs.

3 PROBLEM FORMULATION

We assume that there is an inaccessible MDP underlying the non-Markovian decision problem faced by the agent. Let the state set of the underlying MDP be $\mathcal{S} = \{s^1, s^2, s^3, \ldots, s^N\}$. Let the set of actions available in each state be denoted $\mathcal{A}$. The probability of a transition to state s' on executing action a in state s is denoted $P^a(s, s')$. Note that this transition probability is independent of the states prior to reaching state s (the Markov assumption). The expected value of the payoff received on executing action a in state s is denoted $R^a(s)$. The actions the agent executes constitute its control policy. The task for the learning architecture is to determine a control policy that maximizes the expected value of the infinite-horizon sum of discounted payoffs received by the agent. A discount factor $0 < \gamma < 1$ allows the payoffs distant in time to be weighted less than the more immediate payoffs. Such a policy is called an optimal policy. It is known that for every finite MDP there exists a stationary deterministic policy, $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ that is optimal (see Ross, 1983). Therefore, in MDPs the agent can restrict its search to the finite set of stationary deterministic policies.

In N-MDPs the control agent has sensors that return some estimate of the state of the environment. In particular we will assume that the estimates are elements of $\mathcal{X} = \{X^1, X^2, X^3, \ldots, X^M\}$, where $0 < M$. When the underlying, non-observable MDP is in state s, the sensor reading, or *observation*, is X with fixed probability $P(X|s)$. Note that $P(X|s)$ is independent of the agent's policy. Such an N-MDP is called a partially observable MDP, or POMDP (e.g., Sondik, 1978). In this paper we will consider only POMDPs. Henceforth we will use the word state to refer to an element of the set $\mathcal{S}$, and the word observation to refer to an element of the set $\mathcal{X}$.

In this paper we will prove negative results by giving examples from a subclass of POMDPs that have the special property that the observations are labels for disjoint partitions of the underlying state space $\mathcal{S}$, i.e., $P(X_i|s) = 0$ for all $s \notin S_i \subset \mathcal{S}$, and $P(X_i|s) = 1$ for all $s \in S_i$. In a pictorial representation of such a POMDP (see Figures 1 to 4), an ellipse around a set of states will be used to represent the fact that the enclosed states belong to the same observation.

3.1 STOCHASTIC POLICIES

Consider the possible loss in performance when one applies a conventional RL algorithm developed for MDPs to a POMDP.

Fact 1: Just confounding two states of an MDP can lead to an arbitrarily high absolute loss in the *return* or cumulative infinite-horizon discounted payoff.

Proof: Figure 1 presents a POMDP with two states, one observation, and two actions. The optimal policy in the underlying MDP returns a payoff of R at each time step. Therefore the optimal return in the underlying MDP is $\frac{R}{1-\gamma}$. At best the RL algorithm applied to a POMDP will find the best deterministic memory-less policy.[1] In the POMDP shown in Figure 1 there are only two deterministic policies, because there is only one observation and two actions. In the best case, the agent can get a payoff of R followed by

[1] We conjecture that in general Q-learning will not find the best deterministic memory-less policy. Unlike the Markov case, the Q-values found by Q-learning in POMDPs will depend on the control policy followed during learning. Therefore it may be difficult to make any general statements about Q-learning without restricting the learning policy (see Section 5.1).

an infinite sequence of $-R$'s. Therefore the best return for a deterministic memory-less policy in the POMDP of Figure 1 is $R - \frac{\gamma R}{1-\gamma}$. The loss, $\frac{2\gamma R}{1-\gamma}$, can be made arbitrarily high by increasing R.

□

Fact 1 shows that RL algorithms do not degrade gracefully with the degree of non-Markovianness in a POMDP. We now show that the guarantee of a deterministic optimal policy does not hold for POMDPs.

Fact 2: In a POMDP the best stationary stochastic policy *can* be arbitrarily better than the best stationary deterministic policy.

Proof: Figure 1 shows a POMDP with two states, one observation, and two actions, that has a stationary stochastic policy which beats all stationary deterministic policies. As noted before, the best stationary deterministic policy can at best return a payoff of R followed by a infinite sequence of $-R$'s. The stationary stochastic policy that picks action A with probability 0.5 and action B with probability 0.5, gets an expected payoff of 0.0 at every time step. The resulting increase in the return, $\frac{R(2\gamma-1)}{1-\gamma}$, can be made arbitrarily high by increasing R and setting $\gamma > 0.5$.

□

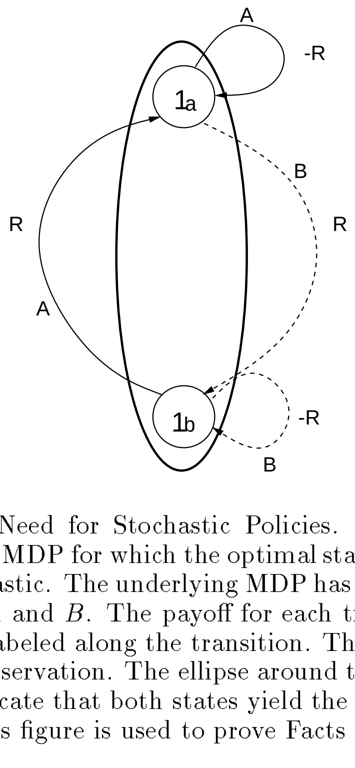

Figure 1: Need for Stochastic Policies. This figure shows a POMDP for which the optimal stationary policy is stochastic. The underlying MDP has 2 states and 2 actions A and B. The payoff for each transition, R or $-R$, is labeled along the transition. The agent sees only one observation. The ellipse around the states 1a and 1b indicate that both states yield the same observation. This figure is used to prove Facts 1 to 4.

Therefore, we propose learning control in the space of stationary stochastic memory-less policies as an alternative to the state-estimation based approaches for dealing with POMDPs. As shown above by example, in POMDPs the best stochastic policy can be significantly better than the best deterministic policy. However, the following fact shows that despite expanding the search space to stationary stochastic policies, one still has to pay a cost for having hidden state.

Fact 3: The best stationary stochastic policy in a POMDP can be arbitrarily worse than the optimal policy in the underlying MDP.

Proof: In Figure 1 the best deterministic policy in the underlying MDP would yield a return of $\frac{R}{1-\gamma}$ As noted before, the best stationary stochastic policy would yield a expected return of 0.0. The difference can be made arbitrarily high by increasing R.

□

Fact 4: In POMDPs the optimal policies *can* be non-stationary.

Proof: Figure 1 shows that the non-stationary policy that picks actions A and B alternately will at worst return a payoff of $-R$ followed by an infinite sequence of R's which is significantly better than the best stationary policy (for $\gamma > 0.5$). Again, the difference can be made arbitrarily large by increasing R.

□

However, searching in the space of non-stationary policies could be prohibitively expensive. Besides, a non-stationary policy is precluded by our intention of learning memory-less policies because a non-stationary policy requires the memory of time elapsed. If memory were allowed, all kinds of memory-based state-estimation techniques could be included and then it is not clear whether there is any advantage to be gained by learning non-stationary policies. Indeed, the proof of Fact 4 also shows that the optimal non-stationary policy in POMDPs can be arbitrarily worse than the optimal memory-less policy in the underlying MDP.

In the rest of this paper, a control policy, π, assigns to each observation a probability distribution over actions. The conventional deterministic policies are a special case of stochastic policies. All policies referred to in this paper will be assumed stationary unless otherwise stated. We will use the symbol Π to denote the space of stochastic policies defined on the observation space of a POMDP. Note that in general Π is not equal to the set of stochastic policies defined over the state space of the underlying MDP. The *return* for, or the value of, a fixed policy $\pi \in \Pi$ in POMDPs is defined as the expected value of the infinite-horizon sum of discounted payoffs, just as in the case of MDPs.

Assumption 1: Throughout the rest of this paper we will assume POMDPs that have the property that the underlying MDPs are ergodic for every stationary policy.

Note that Facts 1 to 4 are also true for ergodic POMDPs. This can be seen by modifying the POMDP in Figure 1 and making it ergodic by adding an $\epsilon > 0$

probability transition from state $1a$ to state $1b$ for action A, and from state $1b$ to state $1a$ for action B. The probability of the self-loops will have to be reduced by a corresponding ϵ. The quantity ϵ can be made small enough to ensure that the modifications have a negligible effect on the returns.

4 EVALUATING A FIXED POLICY

In the Markov case, the value of executing policy π when the starting state of the environment is s, is $V^\pi(s) = E^\pi\{R^{a_0}(s_0)+\gamma R^{a_1}(s_1)+\gamma^2 R^{a_2}(s_2)+\ldots|s_0 = s\}$, where s_i and a_i are the state and action at time step i, and E^π is the expectation symbol under the assumption that action a_i is chosen according to the probability distribution $\pi(s_i)$. Using the Markov assumption, the value of state s under policy π can also be written recursively as follows:

$$\begin{aligned} V^\pi(s) &= \sum_{a\in\mathcal{A}} Pr(a|\pi,s)\left[R^a(s)\right. \\ &\qquad \left. +\gamma\sum_{s'\in\mathcal{S}} P^a(ss')V^\pi(s')\right] \end{aligned} \qquad (1)$$

In a POMDP the value of an observation X under policy $\pi \in \Pi$ cannot be defined in a form similar to Equation 1. However, note that the value of a state s in the underlying MDP does not change just because it is inaccessible. If at any time step the environment enters state s the expected value of the subsequent discounted sequence of payoffs is still $V^\pi(s)$. Therefore we propose that a suitable definition of the value of observation X under policy π is as follows:

$$V^\pi(X) = \sum_{s\in\mathcal{S}} P^\pi(s|X)V^\pi(s) \qquad (2)$$

where $P^\pi(s|X)$ is the asymptotic *occupancy* probability distribution, i.e., the probability that the state of the underlying MDP is s when the observation is known to be X. Note that Equation 2 is only a definition of $V^\pi(X)$, and not an algorithm, because the state's s are not observable in POMDPs.

The asymptotic occupancy distribution can be defined as follows:

$$P^\pi(s|X) = \frac{P(X|s)P^\pi(s)}{P^\pi(X)} = \frac{P(X|s)P^\pi(s)}{\sum_{s'\in\mathcal{S}} P(X|s')P^\pi(s')}$$

where $P^\pi(s)$ is the limiting distribution over the underlying state space of the MDP, and is well defined under Assumption 1.

4.1 WHAT DOES TD(0) LEARN?

Sutton's (1988) TD(0) algorithm is a RL algorithm that is commonly used to evaluate a policy. It is an iterative stochastic approximation algorithm that does not require knowledge of the MDP's transition probabilities, and takes the following form in Markov problems:

$$V_{k+1}(s_k) = (1-\alpha(s_k))V_k(s_k) + \alpha(s_k)(R_k + \gamma V_k(s_{k+1}))$$

where $V_k(s)$ is the k^{th} estimate of $V^\pi(s)$, s_k and R_k are the state and payoff at step k, and α is the learning rate. In Markov problems, under certain conditions on α, TD(0) will converge with probability one to V^π, even if the policy π is stochastic. When applied to a non-Markov problem TD(0) will take the following form:

$$\begin{aligned} V_{k+1}(X_k) &= (1-\alpha(X_k))V_k(X_k) \\ &\qquad +\alpha(X_k)(R_k + \gamma V_k(X_{k+1})). \end{aligned}$$

Theorem 1: In a POMDP of the type defined above, Sutton's TD(0) algorithm will converge to the solution of the following system of equations with probability one (under conditions identical to those required for convergence of TD(0) in MDPs, plus the condition that the learning rates, α, are non-increasing): $\forall\, X \in \mathcal{X}$,

$$\begin{aligned} V(X) &= \sum_{s\in\mathcal{S}} P^\pi(s|X)\left[R^\pi(s)\right. \\ &\qquad \left. +\gamma\sum_{X'\in\mathcal{X}} P^\pi(s,X')V(X')\right], \end{aligned} \qquad (3)$$

where $P^\pi(s,X') = \sum_{s'}(P^\pi(s,s')P(X'|s'))$.

Proof: Consider a semi-batch version of TD(0) that collects the changes to the value function for M steps before making the change. By making M large enough the states of the underlying MDP can be sampled with a frequency that matches $P^\pi(s|X)$ to within ϵ with probability $1-\epsilon$. In Appendix A.1 we prove convergence of the semi-batch TD(0) algorithm outlined above to the solution of Equation 3 with probability one. The semi-batch proof can be extended to on-line TD(0) by using the analysis developed in Theorem 3 of Jaakkola *et al.* (1994). In brief, it can be shown that the difference caused by the on-line updating vanishes in the limit thereby forcing semi-batch TD(0) and on-line TD(0) to be equal asymptotically. The use of the analysis in Theorem 3 from Jaakkola *et al.* (1994) requires that the learning rate parameters α are such that the $\frac{\alpha_t(X)}{max_{t\in M_k}\alpha_t(X)} \rightarrow 1$ uniformly w.p.1.; M_k is the k^{th} batch of size M. If $\alpha_t(X)$ is non-increasing in addition to satisfying the conventional TD(0) conditions, then it will also meet the above "asymptotic flatness" requirement.

□

In general the solution to Equation 3 will not equal the desired value function as defined in Equation 2. Figure 2 from Sutton (1994) presents an example that illustrates a crucial difference between the value function found by the TD(0) algorithm and the correct

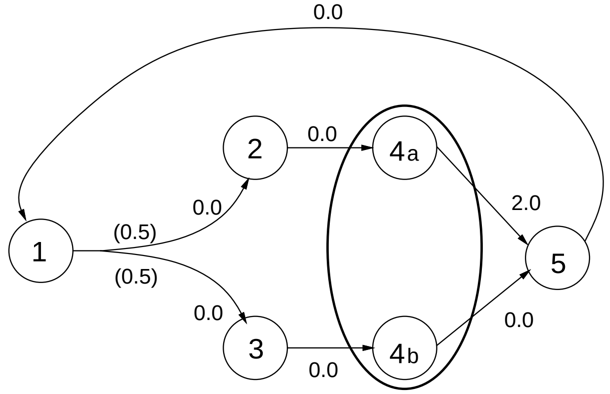

Figure 2: TD(0) and Hidden State. This figure shows a POMDP in which it is clear that TD(0) cannot learn the desired values. States $4a$ and $4b$ are in the same observation. TD(0) will learn a value function that assigns the same value to observations 2 and 3 because both lead to observation 4 with a zero payoff. Therefore, the fact that observation 2 leads reliably to a payoff of two with a delay of one time step while observation 3 does not, will not be discerned by TD(0).

value function (as defined by Equation 2). It shows a six state, five observation POMDP. TD(0), or for that matter any 1-step Markov algorithm will assign the same value to observations 2 and 3 because both lead deterministically into observation 4 with an immediate payoff of zero. The true value function of observation 2, however, will be higher than the true value function of observation 3 because observation 2 reliably leads to a payoff of two after a delay of one time step, while observation 3 does not.[2]

5 OPTIMAL CONTROL

5.1 WHAT DOES Q-LEARNING LEARN?

Q-learning (Watkins, 1989) is a RL algorithm for finding optimal policies in MDPs. One of the big advantages of Q-learning is that it separates exploration from control. In short, the control policy followed during learning has no impact on asymptotic convergence as long as every action gets executed in every state infinitely often. No algorithm for POMDPs can retain that advantage because the control policy followed during learning will impact the occupancy probabilities that are a part of the definition of the return from a policy. To make analysis possible, consider the special case of applying Q-learning with a fixed stationary *persistent excitation* learning policy, i.e., a policy that assigns a non-zero probability to every action in every state, and for which the underlying Markov chain is ergodic. Note that for POMDPs that satisfy Assumption 1 all stationary policies that assign a non-zero probability to every action in every state are persistently exciting. Following such a policy during learning would satisfy the conditions required for $w.p.1$ convergence of Q-learning in MDPs.

Theorem 2: In a POMDP of the type defined above, if a persistent excitation policy π is followed during learning, the Q-learning algorithm will converge to the solution of the following system of equations with probability one (under the same conditions required for convergence of Q-learning in MDPs, plus the condition that the learning rates, α, are non-increasing): $\forall\, X \in \mathcal{X}$,

$$Q(X,a) = \sum_{s\in\mathcal{S}} P^{\pi}(s|X,a)\left[R^a(s) + \gamma \sum_{X'\in\mathcal{X}} P^a(s,X') \max_{a'\in\mathcal{A}} Q(X',a')\right] \tag{4}$$

where $P^{\pi}(s|X,a)$ is the asymptotic probability, under policy π, that the underlying state is s given that the observation-action pair is (X,a), and $P^a(s,X') = \sum_{s'}(P^a(s,s')P(X'|s'))$.

Proof: The proof for Theorem 2 is very similar to the proof of Theorem 1. As in the case of TD(0) consider the semi-batch version of Q-learning that collects the changes to the value function for M steps before making the change. By making M large enough the states of the underlying MDP can be sampled with a frequency that matches $P^{\pi}(s|X,a)$ to within ϵ with probability $1-\epsilon$. In Appendix A we prove that the semi-batch version of Q-learning outlined above converges to the solution of Equation 4 with probability one. The semi-batch proof can be extended to on-line Q-learning by using the analysis developed in Theorem 3 of Jaakkola *et al.* (1994) in a manner similar to that used in the proof of Theorem 1.

□

The solution to Equation 4 suffers from the same problem as the solution to Equation 3 because Q-learning is also based on the 1-step Markov assumption. An additional problem with Q-learning is that it is based on the assumption that a deterministic policy is being sought. An interesting Q-value like quantity can be defined in POMDPs as follows:

$$Q^{\pi}(X,\pi') = \sum_{s\in\mathcal{S}} P^{\pi}(s|X)Q^{\pi}(s,\pi') \tag{5}$$

where $Q^{\pi}(s,\pi') = R^{\pi'}(s) + \gamma \sum_{s'\in\mathcal{S}} P^{\pi'}(s,s')V^{\pi}(s')$.

[2] Unlike TD(0), the more general family of TD($\lambda > 0$) (Sutton, 1988) algorithms average multi-step predictions and will therefore learn a value function that assigns a higher value to observation 2 than observation 3 in the POMDP defined in Figure 2. However, for $\lambda < 1$, TD(λ) will still not be able to learn the value function defined by Equation 2. For POMDPs with absorbing goal states, off-line TD(1), which is equivalent to the Monte Carlo algorithm that averages path-payoffs, will however find the desired value function.

Note that $Q^{\pi}(X, \pi')$ is the Q-value for a stochastic action $\pi'(X)$.

5.2 WHAT IS AN OPTIMAL POLICY?

In discounted MDPs, an optimal policy is simply one that maximizes the value of each state simultaneously. Unfortunately, in *discounted* POMDPs it is no longer possible to define optimal policies in a similar way.

Fact 5**:** In the class of POMDPs defined in Section 3, there need not be a stationary policy that maximizes the value of each observation simultaneously.

Proof: Figure 3 shows a four state, three observation POMDP with two actions A and B. The only policy decision is made in observation 1. Increasing the probability of choosing action A in observation 1 increases the value of observation 1 and decreases the value of observation 2. Increasing the probability of choosing action B in observation 1 has the opposite effect. Therefore, with hidden state there may not be a policy that maximizes the value of each observation simultaneously.

□

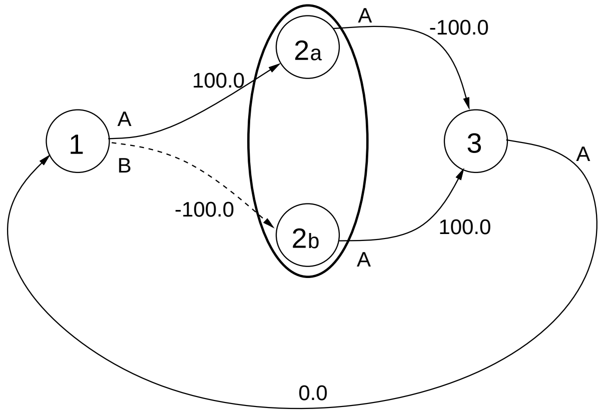

Figure 3: No Policy that Maximizes the Value of Each Observation. This example shows that in general there need not be a policy that simultaneously maximizes the value of each observation. This figure shows a 4 state, 3 observation POMDP. The only policy decision is made in observation 1. Increasing the probability of picking action A increases the value of observation 1 and decreases the value of observation 2. Decreasing the probability of picking action A has the opposite effect.

However, one could imagine that there might be a policy in Π that would simultaneously maximize the value of each state, were they accessible. Even that is not true.

Fact 6**:** In the class of POMDPs defined in Section 3, there need not be a stationary policy that maximizes the value of each *state* in the underlying MDP simultaneously.

Proof: Figure 4 shows a four state, three observation POMDP with two actions A and B. The only policy decision is made in observation 2. The value of state $2a$ inside observation 2 is maximized when action A is chosen with probability one, while the value of state $2b$ is maximized when action B is chosen with probability one.

□

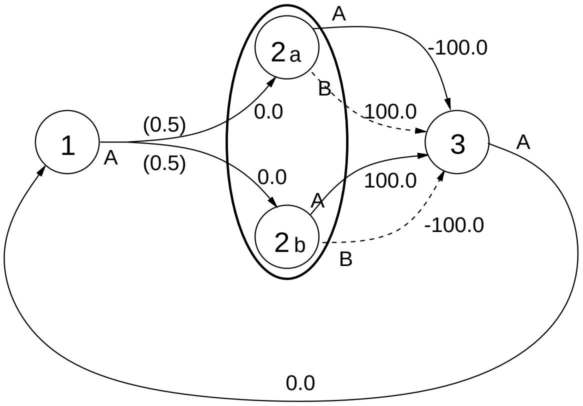

Figure 4: No Policy that Maximizes the Value of Each State. This figure shows a POMDP in which no policy maximizes the value of each state in the underlying MDP simultaneously. The transition probabilities are shown in parenthesis. States $2a$ and $2b$ are in the same observation. There are two actions in observation 2. There is no setting of the policy for observation 2 that simultaneously maximizes the value of states $2a$ and $2b$.

The difficulty in defining an optimal policy in discounted POMDPs can be explained with the help of Equation 2. Changing a policy not only changes the value of each state in the underlying MDP, but also the occupancy distribution of states for each observation. This dual effect makes it possible to trade off the value of one observation with the values of other observations.

5.3 DEFINING AN OPTIMAL POLICY

Discounted Payoff POMDPs: The evaluation of a policy is a vector of values, one value for each state, and as shown above, in general, it is not possible to maximize each element of the value vector simultaneously. One way to overcome that problem is to convert the vector of values into a scalar, e.g., by defining the value of a policy π as $\sum_{X \in \mathcal{X}} P_X V^{\pi}(X)$, where P_X is some weight or measure of the importance of observation X. Some obvious choices for P_X are 1) the probability of starting in observation X, and 2) the probability of occupying observation X. The first option pays undue attention to the starting observation in an infinite-horizon problem. The second option, defines $\pi^* = \arg\max_{\pi\in\Pi} \sum_{X\in\mathcal{X}} P^\pi(X)V^\pi(X)$, and is shown in Fact 7 to be equivalent to maximizing the "average payoff per time step" criterion that is discussed in the next paragraph. Other choices may exist for P_X, but they are unlikely to be reasonable for all POMDPs.

Average Payoff POMDPs: A second policy evaluation criterion that has been studied in the MDP literature (e.g., Bertsekas, 1987) and more recently in the RL literature (Schwartz, 1993; Singh, 1994) is the average payoff per time step criterion. The average payoff under policy π is defined as $\lim_{N\to\infty} E^\pi\{\frac{\sum_{t=0}^N R_t}{N}\}$ and is known to be independent of the starting state for MDPs that are ergodic for all stationary policies. The average payoff is a bounded scalar, and a policy that achieves the maximum value is an optimal policy. The average payoff per time step for a policy in Π is unaffected by the agent's inability to sense the state of the environment.

Let the average payoff for policy π be denoted Λ^π. The *relative* value function in average payoff MDPs is defined as follows (see Bertsekas, 1987):

$$\begin{aligned} V^\pi(s) &= \sum_{a\in\mathcal{A}} Pr(a|\pi,s)\left[(R^a(s)-\Lambda^\pi)\right. \\ &\quad \left. + \sum_{s'\in\mathcal{S}} P^a(s,s')V^\pi(s')\right]. \end{aligned}$$

From here onwards we will use a subscript of γ to distinguish the value function for a discounted decision problem. The definition of the relative value of an observation in an average payoff POMDP is the same as for a discounted payoff POMDP; $V^\pi(X) = \sum_{s\in\mathcal{S}} P^\pi(s|X)V^\pi(s)$. An optimal policy $\pi^* = \arg\max_{\pi\in\Pi} \Lambda^\pi$.

Fact 7: Let V_γ^π be the value function (as defined by Equation 2) for a given POMDP with a discount factor of γ. For the same POMDP, let Λ^π be the average payoff per time step for policy π (without the discount factor). Then, for each $\pi \in \Pi$, $\sum_{X\in\mathcal{X}} P^\pi(X)V_\gamma^\pi(X) = \frac{\Lambda^\pi}{1-\gamma}$. Therefore maximizing $\sum_{X\in\mathcal{X}} P^\pi(X)V_\gamma^\pi(X)$ is equivalent to maximizing the average payoff per time step.

Proof: By definition $V_\gamma^\pi(X) = \sum_{s\in\mathcal{S}} P^\pi(s|X)V_\gamma^\pi(s)$. Therefore,

$$\begin{aligned} \sum_{X\in\mathcal{X}} P^\pi(X)V_\gamma^\pi(X) &= \sum_{X\in\mathcal{X}} P^\pi(X)\sum_s P^\pi(s|X)V_\gamma^\pi(s) \\ &= \sum_s\sum_X P^\pi(X)P^\pi(s|X)V_\gamma^\pi(s) \\ &= \sum_s P^\pi(s)V_\gamma^\pi(s) \\ &= \sum_s P^\pi(s)R^\pi(s) + \gamma\sum_s P^\pi(s) \\ &\quad \sum_{s'} P^\pi(s,s')V_\gamma^\pi(s') \\ &= \Lambda^\pi + \gamma\sum_{s'} P^\pi(s')V_\gamma^\pi(s') \\ &= \Lambda^\pi + \gamma\sum_X P^\pi(X)V_\gamma^\pi(X). \end{aligned}$$

□

6 DISCUSSION

In this paper, we developed a new framework for learning without state estimation in POMDPs by including stochastic policies in the search space and by defining the value of an observation under a given policy. It was demonstrated that the return for a memory-less stochastic policy can be significantly better than the return for any memory-less deterministic policy. However, it should be pointed out that the definition of an optimal policy suggested in this paper is somewhat arbitrary because the only reason to restrict the search space to stationary policies is computational economics.

Note that RL researchers (Sutton, 1990) and learning automata researchers (e.g., Narendra and Thathachar,1974; Barto and Anandan, 1985) have used stochastic policies in the past, but as intermediate policies to ensure sufficient exploration, and always with the view that the ultimate goal is to learn the best deterministic policy. However, researchers in game theory have studied zero-sum games where the optimal strategies are stochastic for the same reason that motivated the search for stochastic policies in POMDPs: the lack of knowledge of the opponent's action constitutes hidden state.

Finally, we presented strong reasons why researchers should use the average payoff criterion to formulate problems that have hidden state, because of the difficulty in defining optimal policies with the discounted payoff criterion.

7 Conclusion

The motivation for this study came from the following simple observation: the first-principles definition of the value of a state under a fixed policy does not involve the Markov assumption and can be computed statistically via Monte Carlo evaluation (Barto and Duff, 1994). This means that for any average payoff POMDP, given enough computational resources it is possible to determine the best policy from any finite set of policies with an arbitrarily high degree of confidence. Unfortunately hidden state introduced two complications. First, the Markov assumption no longer holds, and it was the Markov assumption that allowed efficient search of the policy space via conventional RL-based techniques. Second, in moving from deterministic to stochastic policies we have moved from a finite policy space to an infinite policy space. In this paper we developed a framework for assigning values to observations in POMDPs that does not involve the Markov assumption. In a subsequent paper, we present a new Monte Carlo algorithm for solving average payoff POMDPs that can do an efficient search of the infinite stochastic policy space (Jaakkola, Singh, and Jordan, 1994) as defined in this paper.

A Convergence of semi-batch Q-learning

Let $M_k(X,a)$ be the number of times action a was executed in observation X within the k^{th} batch of size M, $n_k(s|X,a)$ be the number of times the actual underlying state was s when the observation-action pair was (X,a), and $n(X,X'|a)$ be the number of times a transition took place from observation X to observation X' given that action a was executed. The persistent excitation policy followed by Q-learning during learning is denoted π. Then the Q-value of (X,a) after the k^{th} batch is given by:

$$\begin{aligned} Q_{k+1}(X,a) &= (1 - M_k(X,a)\alpha_k(X,a))Q_k(X,a) \\ &+ M_k(X,a)\alpha_k(X,a)\left[\sum_s \frac{n(s|X,a)}{M_k(X,a)} r_k^a(s) \right. \\ &\left. + \gamma \sum_{X'} \frac{n(X,X'|a)}{M_k(X,a)} \max_{a'} Q_k(X',a') \right], \end{aligned}$$

where $r_k^a(s)$ is the sample average of the actual payoffs received on executing action a in state s in the k^{th} batch. Assume $\bar{Q}(X,a)$ is the solution to Equation 4. Let

$$\begin{aligned} F_k(X,a) &= \sum_s \frac{n(s|X,a)}{M_k(X,a)} r_k^a(s) \\ &+ \gamma \sum_{X'} \frac{n(X,X'|a)}{M_k(X,a)} \max_{a'} Q_k(X',a') \\ &- \bar{Q}(X,a), \end{aligned}$$

then, if $V_k(X) = \max_a Q_k(X,a)$ and $\bar{V}(X) = \max_a \bar{Q}(X,a)$,

$$\begin{aligned} F_k(X,a) &= \gamma \sum_{X'} \frac{n(X,X'|a)}{M_k(X,a)} [V_k(X') - \bar{V}(X')] \\ &+ \sum_s \left(\frac{n(s|X,a)}{M_k(X,a)} r_k^a(s) - P^\pi(s|X,a)R^a(s)\right) \\ &+ \gamma \sum_{X'} \left(\left(\frac{n(X,X'|a)}{M_k(X,a)} - P^a(X,X'|\pi)\right)\bar{V}(X')\right), \end{aligned}$$

where

$$P^a(X,X'|\pi) = \sum_s P^\pi(s|X,a) \left[\sum_{s'} (P^a(s,s')P(X'|s')) \right].$$

The expected value of $F_k(X,a)$ can be bounded by

$$\begin{aligned} ||E\{F_k(X,a)\}|| &\leq \gamma ||V_k - \bar{V}|| \\ &+ ||E\{\sum_s \left(\frac{n(s|X,a)}{M_k(X,a)} - P^\pi(s|X,a)\right)R^a(s)\}|| \\ &+ \gamma || \sum_{X'} E\{\left(\left(\frac{n(X,X'|a)}{M_k(X,a)} - P^a(X,X'|\pi)\right)\bar{V}(X')\right)\}|| \\ &\leq \gamma ||V_k - \bar{V}|| + C\epsilon_k^M, \end{aligned}$$

where ϵ_k^M is the larger of

$\max_{(s,X,a)} |E\{\frac{n(s|X,a)}{M_k(X,a)}\} - P^\pi(s|X,a)|$, and

$\max_{(X,X',a)} |E\{(\frac{n(X,X'|a)}{M_k(X,a)}\} - P^a(X,X'|\pi))|$.

For any $\epsilon > 0$, $\exists M_\epsilon$ such that $\epsilon_k^{M_\epsilon} < \epsilon$ (because the sample probabilities converge with probability one). The variance of $F_k(X)$ can also be shown to be bounded because the variance of the sample probabilities is bounded (everything else is similar to standard Q-learning for MDPs). Therefore by Theorem 1 of Jaakkola *et al.* (1994), for any $\epsilon > 0$, with probability $(1-\epsilon)$, $Q_k(X,a) \rightarrow Q_\infty(X,a)$, where $|Q_\infty(X,a) - \bar{Q}(X,a)| \leq \bar{C}\epsilon$. Therefore, semi-batch Q-learning converges with probability one.

□

A.1 Convergence of semi-batch TD(0)

The proof of convergence for semi-batch Q-learning can be easily adapted to prove probability one convergence of semi-batch TD(0) to the solution of Equation 3. Set the persistent excitation policy in the proof for Q-learning to the policy being evaluated, and replace $R^a(s)$ by $R^\pi(s)$ and $P^a(X,X'|\pi)$ by $P^\pi(X,X')$. Everything else follows.

□

Acknowledgements

We thank Rich Sutton for extensive comments and many useful discussions. This project was supported in part by a grant from the McDonnell-Pew Foundation, by a grant from ATR Human Information Processing Research Laboratories, by a grant from Siemens Corporation, and by grant N00014-90-J-1942 from the Office of Naval Research. The project was also supported by NSF grant ASC-9217041 in support of the Center for Biological and Computational Learning at MIT, including funds provided by DARPA under the HPCC program. Michael I. Jordan is a NSF Presidential Young Investigator.

References

Barto, A. G. & Anandan, P. (1985). Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics, 15,* 360–375.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (to appear). Learning to act using real-time dynamic programming. *Artificial Intelligence.* also, University of Massachusetts, Amherst, CMPSCI Technical Report 93-02.

Barto, A. G. & Duff, M. (1994). Monte carlo matrix inversion and reinforcement learning. In *Advances in Neural Information Processing Systems 6*, San Mateo, CA. Morgan Kaufmann.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE SMC*, *13*, 835–846.

Barto, A. G., Sutton, R. S., & Watkins, C. (1990). Sequential decision problems and neural networks. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems 2*, pages 686–693, San Mateo, CA. Morgan Kaufmann.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice-Hall.

Chrisman, L. Planning for closed-loop execution using partially observable markovian decision processes. Submitted to AAAI, 1992.

Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *AAAI-92.*

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning*, *8*(3/4), 341–362.

Jaakkola, T., Jordan, M. I., & Singh, S. P. (1994). Stochastic convergence of iterative DP algorithms. In *Advances in Neural Information Processing Systems 6.* also submitted to Neural Computation.

Jaakkola, T., Singh, S. P., & Jordan, M. I. (1994). Monte Carlo reinforcement learning in non-markovian decision problems. Submitted to COLT94.

Lin, L. J. & Mitchell, T. M. (1992). Reinforcement learning with hidden states. In *In Proceedings of the Second International Conference on Simulation of Adaptive Behavior: From Animals to Animats.*

McCallum, R. A. (1993). Overcoming incomplete perception with utile distinction memory. In Utgoff, P. (Ed.), *Machine Learning: Proceedings of the Tenth International Conference*, pages 190–196. Morgan Kaufmann.

Narendra, K. S. & Thathachar, M. A. L. (1974). Learning automata—A survey. *IEEE Transactions on Systems, Man, and Cybernetics*, *4*, 323–334.

Ross, S. (1983). *Introduction to Stochastic Dynamic Programming.* New York: Academic Press.

Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the Tenth Machine Learning Conference.*

Singh, S. P. (1994). Reinforcement learning algorithms for average-payoff markovian decision processes. In *Proceedings of the Twelth National Conference on Artificial Intelligence*, Seattle,WA.

Sondik, E. J. (1978). The optimal control of partially observable markov processes over the infinite horizon: discounted case. *Operations Research*, *26*, 282–304.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Sutton, R. S. (1990). Integrating architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proc. of the Seventh International Conference on Machine Learning*, pages 216–224, San Mateo, CA. Morgan Kaufmann.

Sutton, R. S. (1994). personal communication.

Tsitsiklis, J. (to appear). Asynchronous stochastic approximation and Q-learning. *Machine Learning.*

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* PhD thesis, Cambridge Univ., Cambridge, England.

Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3/4), 279–292.

Whitehead, S. D. (1992). *Reinforcement Learning for the Adaptive Control of Perception and Action.* PhD thesis, University of Rochester.

Whitehead, S. D. & Ballard, D. H. (1990). Active perception and reinforcement learning. In *Proc. of the Seventh International Conference on Machine Learning*, Austin, TX. M.

Whitehead, S. D. & Lin, L. J. (1993). Reinforcement learning in non-markov environments. working paper.