A Reinforcement Learning Algorithm with Polynomial Interaction Complexity for Only-Costly-Observable MDPs

Roy Fox
Computer Science Department,
Technion IIT, Israel

Moshe Tennenholtz
Faculty of Industrial Engineering and Management,
Technion IIT, Israel

Abstract

An Unobservable MDP (UMDP) is a POMDP in which there are no observations. An Only-Costly-Observable MDP (OCOMDP) is a POMDP which extends an UMDP by allowing a particular costly action which completely observes the state. We introduce UR-MAX, a reinforcement learning algorithm with polynomial interaction complexity for unknown OCOMDPs.

Introduction

An Unobservable Markov Decision Process (UMDP) is a model of an environment with which a decision-making agent is interacting. At each time step, the agent performs an action of its choice, which affects the state of the environment. After each action, the agent obtains a reward which depends on the state reached.

The agent has no sensors, and receives no indication from the environment on the value of its state (hence the name of the model). However, when the model is known to the agent, either partially or completely, it may know the stochastic dependency of the state on its actions. This allows the agent to form a policy which will guide its actions. The quality of the agent's policy is measured by the expected average reward it obtains for a given number of steps (the *horizon*).

UMDPs are important special cases of Partially-Observable MDPs (POMDPs). In POMDPs the agent may possess some (possibly noisy) sensory power, and partially observe the state at each step.

Planning in UMDPs is the problem of calculating an optimal policy for a given UMDP and a given finite horizon. Planning is possible by using, for example, methods of value iteration (Sondik 1971), but was proved to be NP-Complete (Papadimitriou & Tsitsiklis 1987). Learning in UMDPs is the problem of performing near-optimally in a given environment, which implements an unknown UMDP. This Reinforcement Learning task is clearly impossible, because no information of the model can be gathered by the interaction.

An Only-Costly-Observable MDP (OCOMDP) is an extension of an UMDP, in which a particular costly action is added, which allows the agent to completely observe the state. OCOMDPs are also special cases of POMDPs, in which every action provides either complete observability or none at all. In some scenarios, OCOMDPs may be interesting as extensions of UMDPs, which enable learning. For example, a blind robot may be temporarily equipped with an expensive powerful sensor, for the purpose of learning an optimal plan for when it's blind again. As another example, expensive hospital equipment may be used in medical research for learning optimal outpatient treatment. In other scenarios, OCOMDPs may be interesting in their own right.

Planning in OCOMDPs is at least as computationally hard as in UMDPs, since the full-observation action may be too costly to be useful for the given horizon, in which case the problem reduces to planning in the underlying UMDP. Learning in OCOMDPs is obviously at least as computationally hard as planning.

In Reinforcement Learning, however, there's an additional important complexity measure - the number of actions taken by the algorithm, called its *interaction complexity*. To our knowledge, no previously published RL algorithm for partially observable decision processes has a polynomial interaction complexity (for a survey see Hasinoff (2003)).

We introduce UR-MAX, a Reinforcement Learning algorithm for OCOMDPs. It uses polynomial interaction complexity to obtain an expected average reward which is nearly that of the optimal policy for the underlying UMDP. A version of the algorithm competes with the optimal policy for the OCOMDP itself. We also discuss how UR-MAX forms a near-optimal plan for the underlying UMDP, so that in some cases the expensive sensor can eventually be removed without decreased performance.

We note that although full observability is possible in OCOMDPs, algorithms for learning in fully-observable MDPs, such as E^3 (Kearns & Singh 1998) can't simply be extended to OCOMDPs. The full-observation action in OCOMDPs affects not only the agent's payoff, but also the environment, and must be taken prudently. This is not the case, for example, in the formulation of COMDPs presented by Hansen, Barto, & Zilberstein (1997), for which MDP learning algorithms can be adapted.

This paper is organized as follows. In the next section we define the models used and the learning problem. Following, we present the UR-MAX algorithm. Then we analyze the algorithm and prove its near-optimality. Finally, we discuss some extensions and related issues.

Models

Definition 1 (UMDP) *An UMDP is a tuple* $M = \langle S, A, r, t\rangle$ *where*

- S *is the finite set of possible environment states,*
- A *is the finite set of possible agent actions,*
- $r : S \to [0, 1]$ *is the reward function, and*
- $t : S \times A \to \Delta(S)$ *is the transition function, such that for every* $s \in S$ *and* $a \in A$, $t(s, a)$ *is the distribution of the state which follows* s *when* a *is taken.*

The initial state distribution b_0, such that $s_0 \sim b_0$, is known to the agent, and is arbitrary unless defined otherwise.

In an UMDP, the agent only observes its own actions. We allow randomized choices of actions, although they are unnecessary for optimality. The set of observable histories is $H = \bigcup_{i\ge 0} A^i$. The agent's actions are governed by a policy $\pi : H \to \Delta(A)$, where $a_i \sim \pi\left((a_j)_{j=0}^{i-1}\right)$. In particular, $a_0 \sim \pi(\Lambda)$, where Λ is the empty initial history.

The stochastic process $(s_i, a_i)_{i\ge 0}$ is then induced by

- the model M (by having $s_{i+1} \sim t(s_i, a_i)$),
- the initial state distribution b_0, and
- the agent's policy π.

After each step, the agent obtains a reward which is a function of the state reached at the end of the step, but it doesn't observe it. The agent's T-step return is the expected average reward

$$U_M(b_0, \pi, T) = \mathbf{E}\frac{1}{T}\sum_{i=1}^{T} r(s_i).$$

Definition 2 (OCOMDP) *An OCOMDP which extends an UMDP* $M = \langle S, A, r, t\rangle$ *is a tuple* $M' = \langle S, A', r, t', C\rangle$ *where*

- $A' = A \cup \{\alpha\}$ *contains a full-observation action* α,
- $t' : S \times A' \to \Delta(S)$ *extends* t, *and*
- $C \ge 0$ *is the cost of taking* α.

The full-observation action α has two distinct features:

- α allows the agent to observe the state s in which it's taken and its reward $r(s)$. This requires us to redefine the concept of observable histories.
- α costs C to take. This requires us to redefine the agent's return.

After taking α the agent observes the previous state, while after taking some other action it observes nothing. For $i \ge 0$ we define

$$o_i = \begin{cases} s_i, & \text{if } a_i = \alpha; \\ \perp, & \text{otherwise.} \end{cases}$$

The set of observable histories is now redefined to be $H' = \bigcup_{i\ge 0}\left(A' \times (S \cup \{\perp\})\right)^i$, and the agent's policy is now $\pi : H' \to \Delta(A')$, where $a_i \sim \pi\left((a_j, o_j)_{j=0}^{i-1}\right)$.

The agent's T-step return is now

$$U_{M'}(b_0, \pi, T) = \mathbf{E}\frac{1}{T}\sum_{i=1}^{T}\left(r(s_i) - C I_{[a_{i-1}=\alpha]}\right),$$

where I_P is the indicator of the predicate P, being 1 or 0 respectively whether P is true or false. The observation cost is reduced from the reward for each step in which α is used.

In learning, some exploration must precede exploitation, and the agent can't be expected to do as well as in planning. The agent may have to try every action, and go through many state distributions. One implication is that it may take more than T interaction steps to reach a near-optimal T-step return. Another implication is that, by the time the agent forms a near-optimal plan, the state distribution may irrecoverably become arbitrarily worse that the initial one. We must allow our algorithm to compete with the optimal plan for the worst initial state distribution, and we must allow it polynomially many steps to do so.

When learning in an UMDP M with a horizon T and margin ϵ, our goal is therefore to find a policy $\bar{\pi}$ in the extending OCOMDP M' which, when used for $\bar{T} = poly\left(T, \frac{1}{\epsilon}, |M|, C\right)$ steps from any initial distribution b_0, yields a $\bar{T}$-step return of

$$U_{M'}\left(b_0, \bar{\pi}, \bar{T}\right) \ge \min_b \max_\pi U_M(b, \pi, T) - \epsilon.$$

We now turn to define the hypothetical model of a Partially-Known OCOMDP (PK-OCOMDP) used in our algorithm. Similarly to the R-MAX algorithm (Brafman & Tennenholtz 2002), we follow the principle of optimism under uncertainty. The model involves known parts, in which it attempts to approximate the real OCOMDP, and unknown parts, in which it switches to a learning state with an optimistically maximal reward.

Definition 3 (PK-OCOMDP) *For an OCOMDP* $M' = \langle S, A', r, t', C\rangle$, *a PK-OCOMDP is a tuple* $\mathcal{M} = \langle \mathcal{S}, A', \rho, \tau, C, \mathcal{W}, \epsilon'\rangle$ *where*

- $\mathcal{S} = S \cup \{s^*\}$ *contains a special learning state* s^*,
- $\rho : \mathcal{S} \to [0, 1]$ *extends* r *with* $\rho(s^*) = 1$,
- $\tau : S \times A' \to \Delta(\mathcal{S})$ *partially approximates* t' *(the exact meaning of* τ, $\mathcal{W}$ *and* ϵ' *is defined later),*
- $\mathcal{W} = \{\mathcal{W}_a\}_{a\in A'}$ *is a collection of sets of beliefs such that* $\mathcal{W}_a \subseteq \Delta(S)$ *for every* $a \in A'$, *which indicate the part of the model considered known, and*
- $\epsilon' \ge 0$ *is the tolerance of the known boundary.*

The agent's belief in a given model is the distribution of the current state, given what the agent knows, which is the observable history. For every action a, the elements of $\mathcal{W}_a$ are beliefs for which the effect of a has been approximately revealed by the algorithm. The linearity of transitions, described below, allows the agent to deduce the effect of a on beliefs which are linear combinations of elements of $\mathcal{W}_a$.

For every $a \in A'$, we define $Sp(\mathcal{W}_a)$ to be the minimal affine subspace of $\mathbb{R}^S$ which includes $\mathcal{W}_a$

$$Sp(\mathcal{W}_a) = \left\{ \sum_{w\in\mathcal{W}_a} c_w w \;\middle|\; \sum_{w\in\mathcal{W}_a} c_w = 1 \right\}.$$

In particular, $Sp(\emptyset) = \emptyset$.

However, in order to bound our error we must also bound the coefficients in these combinations. For every $a \in A'$ and

$c > 0$, $Sp(\mathcal{W}_a, c) \subseteq Sp(\mathcal{W}_a)$ is the set of linear combinations of $\mathcal{W}_a$ with bounded coefficients

$$Sp(\mathcal{W}_a, c) = \left\{ \sum_w c_w w \in Sp(\mathcal{W}_a) \,|\forall w \in \mathcal{W}_a : c_w \leq c \right\}.$$

Now we allow some tolerance as, for every $a \in A'$, we define $\mathcal{B}_a^{\mathcal{W},\delta}$ to be the set of beliefs which are δ-close to $Sp(\mathcal{W}_a, e)$, in the L_1 metric, where e is the base of the natural logarithm.

$$\mathcal{B}_a^{\mathcal{W},\delta} = \{b \in \Delta(S) | \exists b' \in Sp(\mathcal{W}_a, e) : \|b' - b\|_1 \leq \delta\},$$

where

$$\|b' - b\|_1 = \sum_{s \in S} |b'(s) - b(s)|.$$

$\mathcal{B}_a^{\mathcal{W},\epsilon'}$ represents the set of beliefs, the effect of a on which is considered known while planning. τ is supposed to approximate the real transition function t' for beliefs in $\mathcal{B}^{\mathcal{W},\epsilon'}$.

To define the stochastic process $(s_i, a_i)_{i \geq 0}$ in $\mathcal{M}$, we must consider the transitions of beliefs. In the real model M', the belief which follows b when a is taken is

$$t'_{b,a} = \begin{cases} \sum_s b(s) t'(s, a), & \text{if } a \neq \alpha; \\ t'(o, \alpha), & \text{otherwise,} \end{cases}$$

where $o \sim b$ if $a = \alpha$. Note that, unless $a = \alpha$, the transition is deterministic.

In $\mathcal{M}$, belief-state transitions are defined the same way, except that an action which is unknown for the belief-state from which it is taken always leads deterministically to the special learning state $s^* \in \mathcal{S}$. For every $s \in \mathcal{S}$, let b_s be such that $b_s(s) = 1$. Then the belief which follows $\bar{b} \in \Delta(\mathcal{S})$ when $a \in A'$ is taken is[1]

$$\tau_{\bar{b},a} = \begin{cases} \sum_{s \in S} \bar{b}(s) \tau(s, a), & \text{if } a \neq \alpha \text{ and } \bar{b} \in \mathcal{B}_a^{\mathcal{W},\epsilon'}; \\ \tau(o, \alpha), & \text{if } a = \alpha \text{ and } b_o \in \mathcal{B}_\alpha^{\mathcal{W},\epsilon'}; \\ b_{s^*}, & \text{otherwise.} \end{cases}$$

When $\mathcal{M}$ is used for planning, the process is defined by having, for every $i \geq 0$, $s_i \sim \bar{b}_i$, and $\bar{b}_{i+1} = \tau_{\bar{b}_i, a_i}$, where the observation $o_i = s_i$ is used when $a_i = \alpha$.

When $\mathcal{M}$ is used to simulate the actual interaction with an OCOMDP M', s_i is not really distributed according to the belief in $\mathcal{M}$, but rather the belief in M'. This is a different stochastic process, which may be unknown to the agent, but is still well-defined.

Definition 4 (OCOMDP Approximation) *Let $M' = \langle S, A', r, t', C \rangle$ be an OCOMDP and $\mathcal{M} = \langle \mathcal{S}, A', \rho, \tau, C, \mathcal{W}, \epsilon' \rangle$ a PK-OCOMDP, such that $\mathcal{S} = S \cup \{s^*\}$ and ρ extends r. $\mathcal{M}$ is called a μ-approximation of M' if, for every $0 \leq \delta \leq \epsilon'$, $a \in A'$ and $b \in \mathcal{B}_a^{\mathcal{W},\delta} \subseteq \mathcal{B}_a^{\mathcal{W},\epsilon'}$*

$$\left\| \sum_{s \in S} b(s)(\tau(s, a) - t'(s, a)) \right\|_1 \leq \mu + 2\delta.$$

Note the involvement of the parameter δ, which demonstrates the linear degradation of the transition functions' proximity, as the belief becomes more distant from the span of learned beliefs.

[1]We slightly abuse the notation for the sake of clarity. The beliefs $b \in \Delta(S)$ and $b' \in \Delta(\mathcal{S})$, where $\forall s \in S : b(s) = b'(s)$, are interchangeable in our notation.

UR-MAX

The algorithm operates in T_2 phases of a fixed length T_1, and keeps track of a hypothetical model $\mathcal{M}$. α is only taken in the first step of each phase and when observations are needed for learning.

Actions other than α are taken simultaneously, both actually in the real environment and virtually in the hypothetical one. α actions are first taken in the real environment, and the resulting observation is then simulated in the hypothetical environment. Note that the distribution of observations may be different than the one anticipated by $\mathcal{M}$.

The algorithm classifies phases by their observed initial state. For each initial state σ, it tries to implement an optimal deterministic T_1-step policy π_σ in $\mathcal{M}$ which uses α only in the first step. Occasionally, the algorithm learns something new about $\mathcal{M}$, and updates π_σ.

For every initial state σ and every step of the phase (except for the last one), the algorithm keeps records which attempt to estimate the state distribution after the step. Each record will contain up to K entries. The lack of observations and the determinism of π_σ make these distributions constant.

Let's describe what happens after $\mathcal{M}$ is initialized, and after each time it's subsequently updated. First, the optimal policies π_σ in $\mathcal{M}$ are recalculated. The records are emptied, because the beliefs they represent have changed. Then, in each phase which starts in σ, the algorithm will take some i steps of π_σ, use α to observe the state reached, and add that observation to the record $rec_{\sigma,i}$. i starts at 1, and grows gradually as each sample of size K is completed. This process continues until any of these occur:

- A sample is completed which reveals the previously unknown effects of an action. In this case, the records are used to learn the effects of the action, and $\mathcal{M}$ is updated.
- i reaches T_1, and π_σ is implemented entirely without unknown effects (except maybe in the last step).
- Phases with some other initial state cause $\mathcal{M}$ to be updated, which changes π_σ and empties the records.

As M' becomes more and more known, more phases are completed without requiring an update. Eventually, enough phases are exploiting for the algorithm to achieve near-optimal return.

We are now ready to formally introduce our algorithm. Let $M = \langle S, A, r, t \rangle$ be an UMDP, C a full-observation cost, T a horizon and $\epsilon > 0$ a margin. For convenience, S and r are given as input to the algorithm, but this is unnecessary, as discussed later.

For a record rec, we define $rec(s) = \frac{\#_s rec}{|rec|}$ to be the relative frequency of s among its entries. In other words, $rec \in \Delta(S)$ is the uniform distribution over rec's entries. Whether we refer to the distribution or to the record itself is clear from the context.

The procedures Update and Plan are defined after the main one.

```
UR-MAX$(S, A, r, C, T, \epsilon)$:
1. Initialize:
   • Set $\mathcal{S} = S \cup \{s^*\}$, $A' = A \cup \{\alpha\}$, $n = |S|$, $k = |A|$.
   • Set $\rho(s) = r(s)$ for every $s \in S$, $\rho(s^*) = 1$, and
     – $T_1 = \left\lceil \frac{8(C+2)}{\epsilon T} \right\rceil T + 1$, phases' length,
     – $\epsilon' = \frac{\epsilon}{48 T_1}$, planning tolerance,
     – $\delta = \frac{\epsilon'}{4\sqrt{n} T_1}$, learning tolerance,
     – $\mu' = \frac{\epsilon'}{18 e n T_1}$, approximation error,
     – $K = \left\lceil \frac{n^2}{2\mu'^2} \ln \left( (4n)^3 k T_1 \frac{1}{\epsilon} \ln \frac{1}{\delta} \right) \right\rceil$, sample size, and
     – $T_2 = \left\lceil \frac{(4n)^2 k}{\epsilon} T_1 K \ln \frac{1}{\delta} \right\rceil$, number of phases.
   • Set $\mathcal{M} = \langle \mathcal{S}, A', \rho, \tau, C, \mathcal{W}, \epsilon' \rangle$ to be a PK-OCOMDP
     with $\mathcal{W}_a = \emptyset$ for every $a \in A'$ and arbitrary $\tau$.
   • For every $s \in S$, set $\pi_s$ to Plan$(\mathcal{M}, s, T_1)$.
   • For every $s \in S$ and $1 \le i < T_1$, set $rec_{s,i}$ to be an
     empty record with entries in $S$, and set $rec_{s,0} = b_s$.
2. Repeat for each phase $j$ from 0 to $T_2 - 1$:
2.1. Take $\alpha$, observe the initial state $\sigma$, and set $h = (\alpha, \sigma)$.
2.2. Repeat for each step $i$ from 1 to $T_1 - 1$:
2.2.1. If $|rec_{\sigma,i}| = K$, take $a \sim \pi_\sigma(h)$, and append $(a, \perp)$
       to $h$.
2.2.2. Otherwise:
       • Let $a$ be the last action taken.
       • Take $\alpha$ and add the observed state to $rec_{\sigma,i}$.
       • If now $|rec_{\sigma,i}| = K$ and $rec_{\sigma,i-1} \notin \mathcal{B}_a^{\mathcal{W},\sqrt{n}\delta}$, execute
         Update$(\sigma, i, a)$.
       • Complete phase $j$ by taking $T_1 - 1 - i$ arbitrary non-$\alpha$
         actions, and continue to phase $j + 1$ in line 2.
```

```
Update$(\sigma, i, a)$:
1. Let $f_{rec_{\sigma,i-1},a} = rec_{\sigma,i}$ be the sample gathered. For every
   $w \in \mathcal{W}_a$, $f_{w,a}$ is kept even after the records are emptied.
2. If $\mathcal{W}_a \neq \emptyset$, let $b' = \sum_{w \in \mathcal{W}_a} c_w w$ be $rec_{\sigma,i-1}$'s projec-
   tion on $Sp(\mathcal{W}_a)$.
   • If $\mathcal{W}_a = \emptyset$, or if $\|b' - rec_{\sigma,i-1}\|_2 > \delta$, then let
     $\mathcal{W}'_a = \mathcal{W}_a \cup \{rec_{\sigma,i-1}\}$.
   • Otherwise, there exists some $w' \in \mathcal{W}_a$ such that
     $c_{w'} > e$ (we prove this in lemma 7), and let
     $\mathcal{W}'_a = \mathcal{W}_a \backslash \{w'\} \cup \{rec_{\sigma,i-1}\}$.
3. Find functions $\tau'_{s,a} : S \to \mathbb{R}$, for every $s \in S$, which
   satisfy all of the following linear constraints:
   • For every $s \in S$, $\tau'_{s,a} \in \Delta(S)$.
   • For every $w \in \mathcal{W}'_a$, $\left\| \sum_s w(s) \tau'_{s,a} - f_{w,a} \right\|_1 \le 2\mu'$.
4. If a solution exists:
   • Set $\mathcal{W}_a$ to $\mathcal{W}'_a$.
   • For every $s \in S$, set $\tau(s, a)$ to $\tau'_{s,a}$.
   • For every $s \in S$, set $\pi_s$ to Plan$(\mathcal{M}, s, T_1)$.
5. For every $s \in S$ and $1 \le i < T_1$, empty $rec_{s,i}$.
```

```
Plan$(\mathcal{M}, s, T_1)$:
Using value iteration (Sondik 1971), calculate and output a
policy $\pi$ which maximizes $U_{\mathcal{M}}(b_s, \pi, T_1)$ among determin-
istic policies in which
• the first action is $\alpha$, and
• other then initially, $\alpha$ is never used.
```

Analysis

Theorem 5 (UR-MAX Correctness) *Let $M' = \langle S, A', r, t', C \rangle$ be an OCOMDP which extends an UMDP $M = \langle S, A, r, t \rangle$. Let b_0 be some initial state distribution, T a horizon, and $\epsilon > 0$ a margin. Let $\bar{\pi}$ and $\bar{T}$ be the policy and the number of steps, respectively, used by* UR-MAX$(S, A, r, C, T, \epsilon)$. *Then*

$$\bar{T} = poly\left(T, \frac{1}{\epsilon}, |M|, C\right),$$

and

$$U_{M'}\left(b_0, \bar{\pi}, \bar{T}\right) \ge \min_b \max_\pi U_M(b, \pi, T) - \epsilon.$$

That $\bar{T} = T_1 T_2$ is polynomial is evident from the choice of parameters in line 1 of the main procedure.

Denote by $U^* = \min_b \max_\pi U_M(b, \pi, T)$ the optimal T-step return in M from the worst initial distribution.

In order to prove our main Theorem, we argue that, with high probability, in almost every phase the algorithm implements a near-optimal policy. Namely, we show that, with probability of at least $1 - \frac{\epsilon}{4}$, at least a $1 - \frac{\epsilon}{4}$ fraction of the phases yield a return of at least $U^* - \frac{\epsilon}{4}$.

When π satisfies the restrictions in the Plan procedure, we define $b_i^{\sigma,\pi}$ to be the belief in M' which deterministically results when π is used from b_σ for i steps, and $a_i^{\sigma,\pi}$ to be the action which is deterministically taken in the following step.

A sample is completed when a record $rec_{\sigma,i}$, for some $\sigma \in S$ and $1 \le i < T_1$, reaches size K. Consider the value of $\pi = \pi_\sigma$ at that time. The sample $rec_{\sigma,i}$ is said to have failed if

$$\left\| rec_{\sigma,i-1} - b_{i-1}^{\sigma,\pi} \right\|_1 > \mu',$$

or if

$$\left\| rec_{\sigma,i} - \sum_s rec_{\sigma,i-1}(s) t'(s, a_{i-1}^{\sigma,\pi}) \right\|_1 > 2\mu'.$$

The failure of a sample depends not only on the respective record, but also on the previous one. The algorithm is said to have failed if any sample has failed. Note that, since $b_{i-1}^{\sigma,\pi}$ and t' are unknown to the agent, a failure may pass undetected during the run of the algorithm.

Lemma 6 (Sample Failure) *A sample fails with probability at most $\frac{\epsilon}{(4n)^2 k T_1 \ln \frac{1}{\delta}}$.*

The proof of this lemma is lengthy, and is omitted due to the lack of space. Basically, it is proved by noticing that the entries in $rec_{\sigma,i-1}$ are identically and independently distributed according to $b_{i-1}^{\sigma,\pi_\sigma}$, and those in $rec_{\sigma,i}$ are i.i.d. according to b_i^{σ,π_σ}, and applying Hoeffding's inequality.

For an action $a \in A'$ such that $\mathcal{W}_a \neq \emptyset$, let $d(\mathcal{W}_a) = \dim Sp(\mathcal{W}_a)$ be the dimension of $Sp(\mathcal{W}_a)$, let $H(\mathcal{W}_a) \subseteq Sp(\mathcal{W}_a)$ be the convex hull of $\mathcal{W}_a$, and let $v(\mathcal{W}_a)$ be the $d(\mathcal{W}_a)$-dimensional volume of $H(\mathcal{W}_a)$. For $\delta > 0$, we define

$$F_\delta(\mathcal{W}_a) = d(\mathcal{W}_a) + \ln \frac{d(\mathcal{W}_a)!v(\mathcal{W}_a)}{\delta^{d(\mathcal{W}_a)}} + 1.$$

$F_\delta(\mathcal{W}_a)$ is a measure of how much is known for a, which combines the dimension of $\mathcal{W}_a$ and the volume of its convex hull. For convenience, we define $d(\emptyset) = -1$ and $F_\delta(\emptyset) = 0$.

Lemma 7 (Update Correctness) *After D invocations of the Update procedure, given no failures, $d(\mathcal{W}_a) = |\mathcal{W}_a| - 1$ ($\mathcal{W}_a$ is affinely independent) for every $a \in A'$, $\sum_{a\in A'} F_\delta(\mathcal{W}_a) \geq D$, and $\mathcal{M}$ is a μ-approximation of M', where $\mu = 8en\mu'$.*

Proof (sketch) The proof is by induction on D. For $D = 0$, observe that $\mathcal{M}$ is initialized in line 1 of the main procedure to be a μ-approximation of any real model, that $d(\emptyset) = |\emptyset| - 1$, and that $\sum_a F_\delta(\emptyset) = 0$.

Now we assume that the lemma holds for some $D-1$, and prove it for D. Let the Dth invocation be with parameters σ, i and a. During the invocation, $\mathcal{W}_{a'}$ remains the same for every $a' \neq a$. Consider $rec_{\sigma,i-1}$, b' and $\mathcal{W}'_a$ of line 2 of the Update procedure.

When $\mathcal{W}_a \neq \emptyset$, b' is the projection of $rec_{\sigma,i-1}$ on $Sp(\mathcal{W}_a)$. If $\|b' - rec_{\sigma,i-1}\|_2 > \delta > 0$, i.e. $rec_{\sigma,i-1}$ lies more than δ outside $Sp(\mathcal{W}_a)$, then the dimension of $\mathcal{W}'_a = \mathcal{W}_a \cup \{rec_{\sigma,i-1}\}$ is higher than that of $\mathcal{W}_a$, and the volume of its convex hull is at least $\frac{\delta}{d(\mathcal{W}'_a)}$ times that of $\mathcal{W}_a$.

If, on the other hand,

$$\sqrt{n}\delta \geq \sqrt{n}\,\|b' - rec_{\sigma,i-1}\|_2 \geq \|b' - rec_{\sigma,i-1}\|_1\,,$$

then since $rec_{\sigma,i-1} \notin \mathcal{B}_a^{\mathcal{W},\sqrt{n}\delta}$, $b' = \sum_w c_w w \notin Sp(\mathcal{W}_a, e)$, and there indeed must be some $w' \in \mathcal{W}_a$ such that $c_{w'} > e$. Then $\mathcal{W}'_a = \mathcal{W}_a \backslash \{w'\} \cup \{rec_{\sigma,i-1}\}$ has the same dimension as $\mathcal{W}_a$, and the volume of its convex hull is at least e times larger.

In either of these cases, and also when $\mathcal{W}_a = \emptyset$, $\mathcal{W}'_a$ remains affinely independent, and $F_\delta(\mathcal{W}'_a) \geq F_\delta(\mathcal{W}_a) + 1$.

This update actually takes place in line 4 of the Update procedure, since $\tau'_s = t'(s,a)$, for every $s \in S$, is a solution to the constraint satisfaction problem of line 3, assuming no failures.

Finally, we hint how to show that $\mathcal{M}$ remains a μ-approximation of M' when it's updated. For $w \in \mathcal{W}_a$, $\sum_s w(s)\tau(s,a)$ approximates $\sum_s w(s)t'(s,a)$ because, by the constraints of line 3 of the Update procedure and the assumption of no failures, $f_{w,a}$ approximates both of them.

For any $b' \in Sp(\mathcal{W}_a, e)$, the distance between $\sum_s b'(s)\tau(s,a)$ and $\sum_s b'(s)t'(s,a)$ is a combination of the approximation errors for every $w \in \mathcal{W}_a$. This combination has bounded coefficients, so it can be bounded as well. For any $b \in \Delta(S)$, the approximation degrades linearly with its minimal L_1 distance from any $b' \in Sp(\mathcal{W}_a, e)$, since $\|\sum_s (b'(s) - b(s))t'(s,a)\|_1$ is bounded by that distance, and similarly for τ. ■

Corollary 8 (Update Invocations) *If the algorithm doesn't fail, at most $4nk \ln \frac{1}{\delta}$ invocations of the Update procedure are ever made.*

This stems from the fact that for every $a \in A'$, both the dimension of $\mathcal{W}_a$ and the volume of its convex hull are bounded, so lemma 7 bounds the number of invocations.

Corollary 9 (Exploring Phases) *If the algorithm doesn't fail, at most $\frac{\epsilon}{4}T_2$ entries are ever added to the records.*

Records are bounded in size by K, and by corollary 8 are emptied a bounded number of times.

Corollary 10 (Algorithm Failure) *The algorithm fails with probability at most $\frac{\epsilon}{4}$.*

This is a combination of lemma 6 and corollary 9.

Lemma 11 (Exploiting Phases) *If the algorithm doesn't fail, every phase in which no entries are added to the records is exploiting, i.e. has a return of at least $U^* - \frac{\epsilon}{4}$.*

Proof (sketch) This is a version of the Implicit Explore or Exploit Lemma of R-MAX.

In a phase which starts in σ, in which no entries are added to the records (a non-exploring phase), the records for σ are all of size K. Their distributions are all in $\mathcal{B}_a^{\mathcal{W},\sqrt{n}\delta}$, where a takes the values of their respective actions. Otherwise, an Update would have occurred.

Assuming no failures, the real beliefs in M' are all in $\mathcal{B}_a^{\mathcal{W},\mu'+\sqrt{n}\delta}$. Since $\mathcal{M}$ is a μ-approximation of M', the distance between the beliefs in M' and in $\mathcal{M}$ increases by at most $\mu + 2\mu' + 2\sqrt{n}\delta < \frac{\epsilon'}{T_1}$ at each step, and so the beliefs in $\mathcal{M}$ are indeed all in $\mathcal{B}_a^{\mathcal{W},\epsilon'}$, except maybe in the last step. That means that s^* is not reached during the phase.

Assume a non-exploring phase starts in σ and has policy π. The difference between $U_{M'}(b_\sigma, \pi, T_1)$ and $U_{\mathcal{M}}(b_\sigma, \pi, T_1)$ comes from the difference in beliefs, as well as from the last step of the phase, which in our algorithm may remain unknown indefinitely.

The phase, initial step excluded, can be broken down into $\frac{T_1-1}{T}$ parts of length T. Suppose that an optimal deterministic T-step policy in M is used in each of these parts, and together they form a policy π' in $\mathcal{M}$ of the type required in the Plan procedure (so that $U_{\mathcal{M}}(b_\sigma, \pi, T_1) \geq U_{\mathcal{M}}(b_\sigma, \pi', T_1)$). The difference between U^* and $U_{\mathcal{M}}(b_\sigma, \pi', T_1)$ comes from the difference in beliefs, as well as from the first step of the phase and its cost. Combining these differences, we get

$$U_{M'}(b_\sigma, \pi, T_1) \geq U_{\mathcal{M}}(b_\sigma, \pi, T_1) - \frac{\epsilon'}{2} - \frac{1}{T_1} \geq$$

$$\geq U^* - 6\epsilon' T_1 - \frac{C+2}{T_1} \geq U^* - \frac{\epsilon}{4}.$$

■

Now we can complete the proof of our main Theorem.

Proof (Theorem 5: UR-MAX Correctness) Exploring phases utilize α twice, and may yield as worse a return as $-\frac{2C}{T_1}$. Let p be the expected fraction of phases which are non-exploring. Corollary 9 gives $p \geq 1 - \frac{\epsilon}{4}$, given that the algorithm doesn't fail. Corollary 10 then gives $p \geq (1-\frac{\epsilon}{4})^2$. Assume that $\epsilon < U^* \leq 1$, since otherwise the learning problem is trivial. Then

$$U_{M'}(b_0, \bar{\pi}, \bar{T}) \geq p\left(U^* - \frac{\epsilon}{4}\right) + (1-p)\left(-\frac{2C}{T_1}\right) \geq$$

$$\geq \left(1-\frac{\epsilon}{4}\right)^2 \left(U^* - \frac{\epsilon}{4}\right) - \left(1 - \left(1-\frac{\epsilon}{4}\right)^2\right)\frac{2C}{T_1} >$$

$$> U^* - \frac{\epsilon}{2}U^* - \frac{\epsilon}{4} - \frac{\epsilon C}{T_1} > U^* - \epsilon.$$

■

Discussion

Learning States and Rewards

It's reasonable to demand that the algorithm doesn't receive r as input, but is required to infer it from the environment. For that purpose, we assume that the rewards of states observed by α are visible as well. It can be verified that the rewards of states which were never observed by α play absolutely no role in UR-MAX. These states are missing from the records when learning, and therefore aren't reachable when planning.

The names of the states are of no consequence. $n = |S|$ is, however, required for the calculation of parameters. If n is not given as input, the following algorithm enables learning nonetheless.

Suppose that at some point only $n' < n$ states of M were observed by the algorithm (initially, $n' = 0$). Imagine a model $\bar{M}$ with $n' + 1$ states, in which all the unobserved states of M are replaced with a single state. This state has a reward of 1 and leads deterministically to itself. The probability of reaching the invented state from each observed state is the sum of those probabilities for the unobserved states.

The algorithm runs UR-MAX with $\bar{M}$'s states as input. If new states are revealed, the process is repeated, again and again, with increasing values of n'. Eventually UR-MAX is run without revealing new states. At that point the algorithm is unable to tell if the real model is M or $\bar{M}$. Since UR-MAX competes with the optimal policy of $\bar{M}$, which is at least as rewarding as that of M, it is indeed near-optimal. It should then be continued for some more phases to compensate for the interaction spent with too few states (newly observed states at this point may be ignored).

Competing with OCOMDPs

Since the optimal return of an OCOMDP may after all be larger then that of the UMDP it extends, it's interesting to ask whether a variant of UR-MAX exists which obtains nearly the optimal return of the OCOMDP.

This is possible with two changes of the algorithm. First, the Plan procedure must allow α to be taken anywhere in the phase. Second, the bookkeeping and the analysis must be amended. The beliefs are no longer determined by the initial state of a phase and the number of steps taken in it. Rather, they depend on the last observation, on the number of steps in the phase until it was made, and on the number of steps taken since then.

A Near-Optimal Policy for the UMDP

UR-MAX obtains a near-optimal return, but doesn't actually find a near-optimal policy for the UMDP. It may be desirable to add α to the UMDP only for a bounded learning period, and then to remove it while keeping its insights.

The main issue in finding a near-optimal T-step policy, is identifying the initial state distribution from which the policy will be implemented. UR-MAX may not learn to plan from any initial belief. In fact, in exploiting phases it uses α merely as a reset action, to reach a belief for which a near-optimal policy is known.

If the UMDP had a reset action, it would be possible to start each phase by taking it, instead of α. Then UR-MAX could output a near-optimal policy which begins with the reset action. Moreover, let a connected UMDP be an UMDP in which for every $s, s' \in S$ there exists a sequence of actions which can reach s' from s with positive probability. Then it's possible to use the approximate reset strategy defined by Even-Dar, Kakade, & Mansour (2005), and have UR-MAX produce a near-optimal policy for a connected unknown UMDP which starts with the approximate reset strategy.

Infinite Horizon

Let the infinite-horizon return of a policy be the infimum limit of the returns of its prefixes. In infinite-horizon learning, the agent is required to converge to a return which is nearly the supremum over all policies.

If UR-MAX is run indefinitely, each time with double the horizon and half the margin of the time before, then for every $\epsilon > 0$ it converges to a return within ϵ of the supremum, at a rate which is only polynomially worse than the optimal rate.

References

Brafman, R., and Tennenholtz, M. 2002. R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 213–231.

Even-Dar, E.; Kakade, S.; and Mansour, Y. 2005. Reinforcement learning in POMDPs without resets. *International Joint Conferences on Artificial Intelligence* 690–695.

Hansen, E.; Barto, A.; and Zilberstein, S. 1997. Reinforcement learning for mixed open-loop and closed-loop control. In *Advances in Neural Information Processing Systems*, 1026–1032. MIT Press.

Hasinoff, S. 2003. Reinforcement learning for problems with hidden state.

Kearns, M., and Singh, S. 1998. Near-optimal reinforcement learning in polynomial time. In *Proc. 15th International Conf. on Machine Learning*, 260–268. Morgan Kaufmann, San Francisco, CA.

Papadimitriou, C., and Tsitsiklis, J. 1987. The complexity of markov decision processes. *Mathematics of Operations Research* 441–450.

Sondik, E. 1971. *The Optimal Control of Partially Observable Markov Processes*. Ph.D. Dissertation, Stanford University.