Adapting Reinforcement Learning Algorithms to Episodic Learning of POMDPs
MS&E 338 Final Report
Winter 2014-15

Stephen Ragain
sragain@stanford.edu

Milind Rao
milind@stanford.edu

Abstract

The problem considered in the paper is the joint learning and planning or Reinforcement Learning (RL) problem for Partially Observable Markov Decision Processes (POMDP) with unknown rewards and dynamics. We formulate an episodic learning problem with partial state knowledge. We then adapt UCRL and PSRL into heuristics for this episodic POMDP-RL problem in several ways and apply these heuristics to a adaptations of difficult RL problems into POMDPs. We then present simulation results and analysis.

Formulation

Let $(S, A, P, R, \Omega, \rho, H, L)$ be a POMDP where S is the state space, A the action set, $\mathbb{P}(\cdot|s, a)$ set of conditional probabilities between states, R the reward function, Ω the observation set, H the episode length, and E the number of episodes. For episode $\ell \in \{1, \ldots, L\}$, the initial state is drawn from some $s_0 \sim \rho$ and we observe o_0. For $t \in \{1 \ldots H\}$ we let $s_t^\ell, a_t^\ell, r_t^\ell$ and o_t^ℓ respectively denote the state, action, reward and observation at time t. Each a_t will be computed using a policy $\mu_t^\ell : \Omega \to A$ that we determine at the start of the episode according to our algorithm and using the data from the previous $\ell - 1$ episodes. We assume that the underlying MDP is time-homogeneous.

We will consider problems with a finite E, with the goal

$$\max \mathbb{E}\left[\sum_{\ell=1}^{E}\sum_{t=1}^{H} r_t^\ell\right]$$

In order to make the problem more manageable while still keeping the episodic learning context, we assume that we know the probability distribution of the state given the observation, i.e. we know $O(s|o) = Pr(s_t = s|o_t = o)$. So we are trying to produce optimal behavior in an unknown environment where our ability to measure our state is imperfect, but imperfect in an understood way. It is unclear if we have any hope of learning without making other assumptions.

In the problems we consider in this work, Ω is finite, but some of our adaptations of UCRL and PSRL could be applied to a continuous observation set, provided the state space is still discrete.

Due to the independence of the starting state of an episode from the previous episodes, $\tilde{o}_t^\ell = (o_0^t, a_1^t)$ is a Markov chain with respect to the filtration generated by the first $\ell - 1$ episodes. By augmenting the state space to include all observation-action sequences of length at most H, we obtain a typical reinforcement learning problem. Unfortunately we introduce the curse of dimensionality- the augmented state space will have size $O(|\Omega|^H|A|^H)$, meaning that RL algorithms are likely to be intractable for interesting H. Our hope is that by adapting the RL algorithms to shorter histories or representing history in a condensed state, we can attain strong performance and high tractability.

Related Approaches

Early work on POMDP in RL in [6] focused on defining a Q to $\Omega \times A$ using the empirical P and R and Monte-Carlo estimations, but require knowledge about average rewards and offer no tractability guarantees or experimental results. Work on the "Consistent Representation" method presented in [9] seeks to control a Hidden Markov Model optimally using an internal state space which is Markov, and then applies ϵ-greedy learning methods to the internal space. Similarly there are no tractability or learning rate guarantees for this method, and a generalization of ϵ-greedy approaches seems unlikely to offer good performance because such algorithms learn exponentially in the episode length H on certain classes of RL problems

Belief-Augmentation

The belief state is a sufficient statistic for optimal control [2]. There is a body of recent work [3, 4] that attempts to solve the POMDP RL problem by storing a belief state b_t, a posterior distribution over the states given all history. The belief state contains history implicitly and is updated using Bayes' Rule. Bellman's Equation takes a convenient form for these problems:

$$Q^*(b_t, a) = R(b_t, a) + \sum_{o \in \Omega} P(o_{t+1} = o | b_t, a) V^*(b_{t+1}^{a,o}),$$

where $V^*(b_{t+1}^{a,o})$ is the value of our expected future belief under $a_t = a$ and $o_{t+1} = o$. Computing $V^*(b_{t+1}^{a,o})$ is often intractable. In a later section we show that even in a two-state, two-action problem with known transition and reward dynamics, the exact solution for a POMDP can be intractable for large H.

Ross et al. [8] has an extended survey on Bayesian adaptive POMDP problems and showed that the problem of learning POMDPs can be reduced to a learning MDPs by augmenting the state-space with belief about state and model. They provided theoretical guarantees for approximate approaches which discretize the belief state space. Also in the paper are heuristics to update the belief state which have informed our heuristics in this work.

In [3], the intractability of computing V^* is handled with a point estimator, and the learning problem is explored in a different context where O is not known but there is an oracle/expert that the system can query at any time to get the optimal action. While this work may be effective in implicitly learning complex tasks already solved by other systems, it cannot be applied to a system whose optimal control is unknown.

Tree-Based Algorithms

McCallum describes an algorithm in [5] for learning the underlying state space by using a tree representation for the space of possible histories $\tilde{o}_t^\ell$. Such a representation yields an approximation of the state space as the leaves of this tree. This approach is expanded on in [4] in the context of making Belief-Augmentation more tractable for a traditional RL problem. Rather than storing history implicitly using Bayes' rule as in the formulation above, these approaches try estimate the solutions to the belief update by using the tree structure to represent dependencies.

PSRL Adaptations

Thompson sampling or posterior sampling has been shown to be an effective method at solving RL problems [7]. At the start of each episode, the unknown transition probabilities and rewards are sampled from a prior and the optimal policy for this MDP is implemented through the episode. The rewards and transitions observed for states and actions are then used to update the priors of the reward and transition probabilities for the succeeding time steps.

The challenge in adapting PSRL for learning and planning in POMDPs is twofold :

- Updating the priors for the reward and transition probabilities given partial observations of the state. This could also be restated as the problem of updating counts or statistics used for updating priors given observations from an episode.
- Implementing a policy for partially observed states obtained from the above MDP. This is the POMDP problem.

Similar to POMDP solutions, we maintain an approximate belief state b_t at each time instant of the episode representing the probability of being in a particular state given the model M and past history of observations and actions, i.e. $\mathbb{P}(s_t \mid M, a_1^t, o_1^t) \approx b_t(s_t)$. We use the belief state to update counts for updating priors over the model for the next episode as well as choosing actions in the current episode where we have already sampled a model.

PSRL-POMDP algorithm

The algorithm is described in Alg. and individual parts of the algorithm explained here.

Algorithm 1 PSRL-POMDP

```
for ℓ = 1 : E do
    Sample R_ℓ(s, a), P_ℓ(s_+ | s, a) from priors using counts, rewards, etc.
    Find optimal Q(s, a) for model M'
    for t = 1 : H do
        observe o_t^ℓ
        b_t^ℓ ← UPDATE(b_{t−1}^ℓ, o_t^ℓ, a_{t−1})
        a_t ← OPTACT(b_t^ℓ, Q(·, ·))
    end for
end for
```

Sampling the model

A natural prior to maintain for transition probabilities over the finite state problem is the Dirichlet prior which is parametrized by number of transitions made from an (s, a) tuple to next state s_+. Priors such as Jeffrey's, Beta-Bernoulli or Normal-Gamma depending on prior knowledge are natural priors to use for the stochastic reward for each state and action pair. These are parametrized typically by the number of observations, mean and variance. At the start of every episode, the rewards $R(s, a)$ and transition probabilities $P(s_+ \mid s, a)$ are sampled from the priors to obtain an MDP M'. The optimal solution for the MDP is found through dynamic programming and Q functions obtained.

Choosing action

Given current belief state b_t at each time instant and the Q functions $Q(s, a)$ of sampled model, choosing the action amounts to solving the POMDP problem. While solving the exact problem is not feasible especially for large state and action sequences, some approximate approaches are:

1. Most Likely State :
$$a_t = \operatorname*{argmax}_a Q(\operatorname*{argmax}_s b_t(s), a)$$

2. Sample from belief state :
$$\begin{aligned} s &\sim b_t \\ a_t &= \operatorname*{argmax}_a Q(s, a) \end{aligned}$$

3. Polling :
$$\begin{aligned} \delta_s &= b_t(s) \operatorname*{argmax}_a Q(s, a) \\ a_t &= \operatorname*{argmax}_a \sum_s b_t(s) \mathbf{1}_a(\delta_s) \end{aligned}$$

4. Minimizing expected cost :
$$a_t = \operatorname*{argmax}_a \sum_s b_t(s) Q(s, a)$$

Updating the belief state

Since we do not know the exact transition probabilities, the belief state we are maintaining at each time instant are approximate. Here are some approximate approaches to updating the belief state:

1. Using observation only :
$$b_t(s) \propto \mathbb{P}(o_t \mid s)$$
2. Combining history and observation :
$$b_t(s) \propto \mathbb{P}(o_t \mid s) \sum_{s_-} \mathbb{P}(s \mid s_-, a_{t-1}) b_{t-1}(s_-)$$

Both these methods rely on Bayes' rule for updating the belief state. Apart from just the observation, the latter also relies on prior belief and transition probabilities. Instead of using the sampled model, the expected transition probabilities which are proportional to the transition counts are more appropriately used.

Updating counts

If we are working with the most likely state or a sampled state and choose an action based on that, we can update the transition counts to the most likely (or sampled) state in the next time instant.

Another method to update counts would be to factor in the belief state more explicitly. For instance, updating the transition count can be performed as,

$$n(s, a, s_+) \leftarrow n(s, a, s_+) + b_t(s) b_{t+1}(s_+).$$

The mean reward can be similarly updated as,

$$R_\mu(s, a) \leftarrow \frac{R_\mu(s, a) n(s, a) + b_t(s) r_t}{n(s, a) + b_t(s)},$$

where $n(s, a) = \sum_{s_+} n(s, a, s_+)$.

UCRL Adaptations

Choosing actions

An intuitive extension of UCRL to the POMDP case would be to be optimistic over a range of "reasonable" belief states in addition to the optimistic optimization done over plausible model. Two problems with this approach are that the third layer of optimism is almost always intractable, and that we cannot expect a confidence interval over belief states to contract to a single state in the limit: the state cannot be learned in most practical or interesting POMDPs.

Given that a belief state is a sufficient statistic for POMDPs, however, it makes sense to be optimistic with respect to our current approximation of the belief state $\tilde{b}_t^\ell$. Let $\mathcal{M}^\ell$ denote the set of plausible MDPs given our approximations $\hat{P}^\ell$ and $\hat{R}^\ell$ of the empirical transition dynamics and reward distributions at episode ℓ.Let $\tilde{Q}^\ell$ be the optimistic approximation of Q for episode ℓ:

$$\tilde{Q}^\ell(s, a) = \max_{\tilde{P}, \tilde{R} \in \mathcal{M}^\ell} \tilde{R}(s, a) + \sum_{s'} \tilde{P}(s' | s, a) V(s')$$

where $\tilde{P}$ and $\tilde{R}$ are constrained using the typical confidence set bounds, as in [1], and $V(s') = \max_a \tilde{Q}^\ell(s', a)$. At the start of each episode we will compute this optimistic Q using our past data with the *OptQ* function. Optimizing Q over $\mathcal{M}^\ell$ is tractable because it is a linear program.

As in PSRL, we explore three ways to chose our actions:

$$\hat{s}_t^\ell \leftarrow \operatorname*{argmax}_s Pr(s_t^\ell = s \mid \tilde{b}_t^\ell), \quad a_t^\ell \leftarrow \operatorname*{argmax}_a \tilde{Q}^\ell(\tilde{s}_t^\ell, a) \tag{1}$$

$$a_t^\ell \leftarrow \operatorname*{argmax}_a Pr\left(a = \operatorname*{argmax}_{a'} \tilde{Q}^\ell(s_t^\ell, a') \mid \tilde{b}_t^\ell\right). \tag{2}$$

$$a_t^\ell \leftarrow \operatorname*{argmax}_a \sum_s \tilde{b}_t^\ell(s) \tag{3}$$

In the pseudo code for these algorithms we let $\textsc{OptAct}(b_t^\ell, \tilde{Q}^\ell)$ return the output of one of these procedures.

Updating Statistics

Although UCRL uses statistics (counts, model estimations) differently than PSRL, the choices for updating remain the same- we can choose to update our approximation of the belief state by using only recent information or by repeatedly applying Bayes' rule at each time step, and we can update our counts by acting as if the most likely state were the true state or by using our approximation of the belief state to work in our uncertainty about the true transitions. In the pseudocode below, $\mathcal{D}^\ell$ denotes all relevant data from episode ℓ (belief states, observations, actions, etc.), and UpdateData$(\hat{P}^\ell, \hat{R}^\ell, \mathcal{D}^\ell)$ returns $P^{\ell+1}$ and $R^{\ell+1}$ according to whichever method we have chosen to process the data.

Similarly, we let UpdateBelief$(\tilde{b}^\ell_{t-1}, a^\ell_{t-1}, o^\ell_t)$ return our new estimate of the belief state $\tilde{b}^\ell_{t+1}$ according to whichever method we have chosen for updating our approximate belief state.

Pseudocode

Algorithm 2 UCRL-POMDP

for $\ell = 1 : E$ **do**
  $\tilde{Q}^\ell \leftarrow$ OptQ$(\mathcal{M}^\ell)$
  for $t = 1 : H$ **do**
    observe o^ℓ_{t+1}
    $a^\ell_t \leftarrow$ OPTACT$(b^\ell_{t-1}, \cdot)$
    $b^\ell_t \leftarrow$ UpdateBelief$(b^\ell_{t-1}, a^\ell_t, o^\ell_{t+1}, \cdot)$
    $\mathcal{M}^\ell \leftarrow$ UpdateData$(\mathcal{D}^\ell)$
  end for
end for

Exact solutions

Consider the 2 state POMDP-RL with $S = \{1, 2\}$, $O = S$, $A = \{a, b\}$.

$$\begin{aligned} \mathbb{P}(o_t \mid s_t) &= \delta \mathbf{1}_{[s_t]}(o_t) + (1-\delta)\mathbf{1}_{[S\setminus s_t]}(o_t) \\ \mathbb{P}(s_{t+1} = 2 \mid s_t, a_t) &= p^{a_t}_{s_t} \\ R_t \mid s_t, a_t &\overset{\text{iid}}{\sim} \mathsf{Bern}(r^{a_t}_{s_t}) \end{aligned}$$

Here $p^{\cdot}_{\cdot}, r^{\cdot}_{\cdot}$ are unknown parameters and δ is known.

Let us consider a Bayesian approach to solving the problem. We consider beta-Bernoulli priors for rewards $(r^{a/b}_{1/2,\alpha/\beta})$, location $(l_{1/2})$ and transition probabilities for all actions and states $(p^{a/b}_{1/2,\alpha/\beta})$. The Bellman equation can be written as,

$$\begin{aligned} q_z &= \sum_{\substack{z_1\in\{1,2\},\\ z_2,z_3\in\{\alpha,\beta\}^2,\\ z_4\in\{0,1\}}} \mathcal{P}l(z_1)\mathcal{P}r^z_{z_1}(z_2)\mathcal{P}p^z_{z_1}(z_3)\delta^{z_4}(1-\delta)^{1-z_4} V(r^z_{z_1,z_2}+1, p^z_{z_1,z_3}+1, l_{z_3}\leftarrow l_{z_1}+z_4, l_{\bar{z}_3}\leftarrow l_{\bar{z}_1}) \\ V(r^{\cdot}_{\cdot,\cdot}, p^{\cdot}_{\cdot,\cdot}, l_\cdot) &= \max_{z\in\{a,b\}} q_z, \end{aligned}$$

where operator $\mathcal{P}Z(\alpha) = Z_\alpha/(Z_\alpha + Z_\beta)$ for a beta-bernoulli pair of variables $Z_{\alpha,\beta}$. Here $l_\alpha = l_1$ and $l_\beta = l_2$ and $\bar{1} = \bar{\alpha} = 2 = \beta$. Essentially, given our current priors on transition and reward probabilities and belief about location as states of our value function, we update these priors independently given our expected observation and action. Planning is done based on this. For large state spaces or episode lengths, planning over such a horizon can become intractable.

Results

We tested our heuristics on an example inspired by the *riverSwim* problem or the MDP studied in class. The state space are the integers $1, \ldots, S$ (we use S to denote both the number of states and set of states,

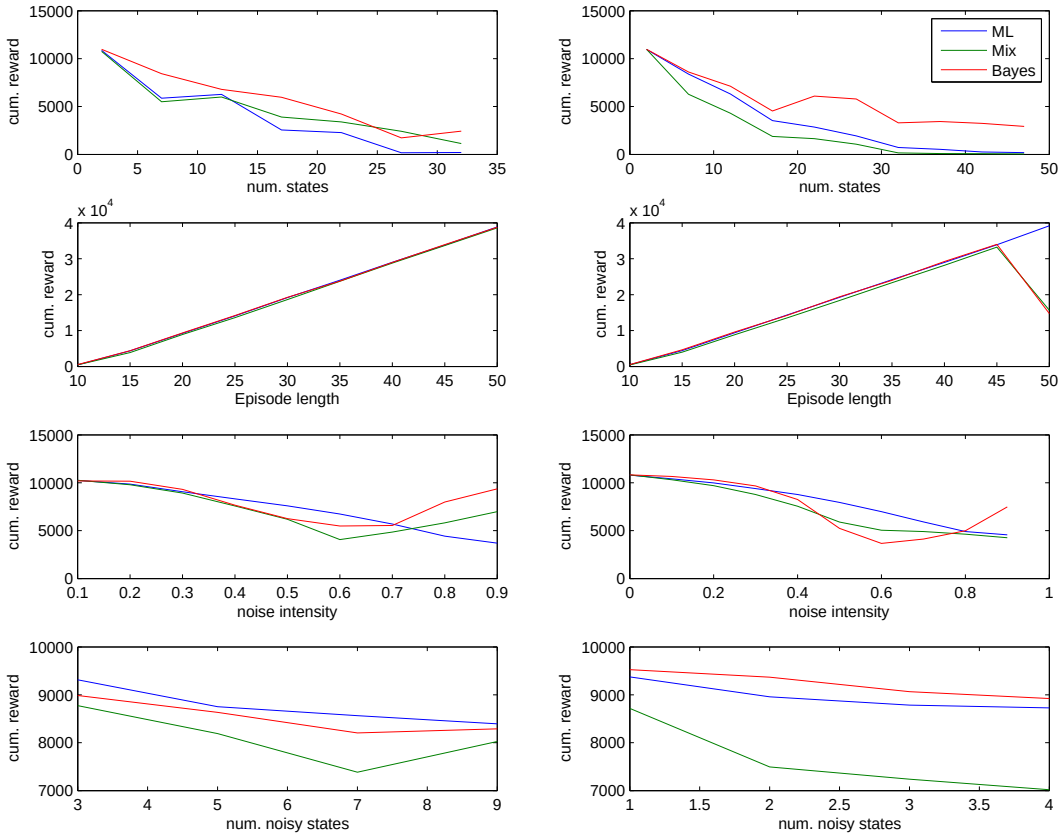

Figure 1: Performance of adaptations with various parameters. On the left side are performance of UCRL algorithms and on the right are PSRL algorithms.

but the distinction should be clear), and there are two actions at each state. At state i, one of the actions is randomly selected to go to state $i+1$ (or S if already at S), and the other leads to $i-1$ (or 1). There is a reward of 1 for any action taken at state S. All other rewards are zero. For an observation error of ϵ, the distribution of observation error (unless varied) was $O(i|s) = 1-\epsilon$, while $O(i+1|s) = O(i-1|s) = \epsilon/2$.

In order to test the performance of our adaptations with respect to different aspects of the POMDP RL problem, we ran several flavors of PSRL-POMDP and UCRL-POMDP while varying

- The number of states : This is a proxy for how the algorithms perform with delayed reward. Notions of planning to learn will be tested here.
- The length of each episode : This is a measure of how much flexibility a system has. If $H = S$, is the system makes a mistake based on faulty observations at any stage, it will not generate reward that episode and has a smaller window to learn.
- The noise "intensity": the probability for which the observation is not the actual state
- The noise shape: The noisy observation is drawn from a wider distribution around the actual state centred at the actual state.

The left column corresponds to UCRL-POMDP and the right column to PSRL-POMDP

For both experiments, "ML" refers to the max likelihood of current state, and refers to using a belief state with a point mass on the most likely state given the most recent observation. "Mix" refers to using only the current observation for a belief state. "Bayes" refers to the full Bayesian update for all belief states and histories.

When held constant relative to the experiment at hand, the parameters were $E = 1000, S = 10, H = S + 10$, and $\epsilon = .3$.

In general the results are relatively consistent between the two approaches, and perform well even compared to the solutions of the underlying MDPs.

In the number of states experiments, we note that optimal rewards at each time period are 11, and for the lower numbers of states, all versions of both algorithms have near optimal control. As the state numbers increase, we see an expected performance drop off because the system becomes more difficult to learn because rewards are at greater delay. The relative flexibility afforded by the larger episode length also drops away and the learner is more susceptible to observation error relative to correct ones in each episode. We see that generally, the bayesian approach outperform the more "myopic" ones, especially in the tail.

In the episode length row, we fix S at 10 and increase H. Thus the optimal control yields rewards of $H - S + 1$, and we see that both adaptations in all of their flavors are performing near optimally even for smaller H, and seem to be obtaining nearly all of the rewards available as H increases.

In the noise intensity row, we vary ϵ, the observation error. This corresponds to putting less weight on the underlying state when drawing the observation. We see a dip in performance around $2/3$, which makes sense considering that the distribution of the observation has maximum entropy at this ϵ. For higher observational errors, the bayesian approaches actually improve their performance, which may signify that they are learning to turn the drop in mass at the actual state as a signal for where that state really is.

Finally, in the row for noise shape, we experiment with increasing the number of possible observations for each state. These probability distributions take on a "triangular" shape with the mode at the underlying state. As expected, as the number of underlying states decreases, performance of all metrics decreases, but surprisingly the Bayesian approaches do not seem to scale better as the noise is spread across the state space.

These results offer insight into when various heuristics can be used. For problems which do not require extensive planning to learn or when observation error is benign, simply using the maximum likelihood estimate is sufficient for updating beliefs and choosing which action to take given observations.

Our results also indicate that it may not be possible to perfectly learn the transition and reward probabilities if observations are noisy.

Conclusions and Future Work

In this work, we adapted UCRL and PSRL algorithms for simultaneous model-based planning and learning in POMDPs. We did this by focussing on various tractable heuristics for updating belief given current observation and history, choosing action based on belief of the model parameters and state belief, and finally updating knowledge of model parameters.

Performance of these heuristics was generally strong relative to performance of PSRL and UCRL on the underlying MDPs with completely observable state. It is unclear to what extent this signifies that successful ideas in RL will extend generally to the POMDP RL setting. Because our example was in many senses "learnable" even with the observation error, it may be that the success of the adaptations came from the nature of the noise that we applied.

Future work on developing theoretical bounds on performance loss as a function of observation error for a class of POMDPs(even if the underlying model is known) may be useful in developing bounds for performance loss in POMDP RL relative to RL on its underlying POMDP. These could help separate the losses from noisy observations confounding the algorithm from what it intends from the losses due to difficulty learning in a noisy environment.

References

[1] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 89–96. Curran Associates, Inc., 2009.

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 3rd edition, 2005.

[3] Finale Doshi, Joelle Pineau, and Nicholas Roy. Reinforcement learning with limited reinforcement: Using bayes risk for active learning in pomdps. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 256–263, New York, NY, USA, 2008. ACM.

[4] Raphael Fonteneau, Lucian Busoniu, and Rémi Munos. Optimistic planning for belief-augmented markov decision processes. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*, pages 77–84. IEEE, 2013.

[5] R Andrew McCallum, G Tesauro, D Touretzky, and T Leen. Instance-based state identification for reinforcement learning. *Advances in Neural Information Processing Systems*, pages 377–384, 1995.

[6] T Michael and I Jordan. Reinforcement learning algorithm for partially observable markov decision problems. *Proceedings of the Advances in Neural Information Processing Systems*, pages 345–352, 1995.

[7] Ian Osband, Dan Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3003–3011. Curran Associates, Inc., 2013.

[8] Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *The Journal of Machine Learning Research*, 12:1729–1770, 2011.

[9] Steven D Whitehead and Long-Ji Lin. Reinforcement learning of non-markov decision processes. *Artificial Intelligence*, 73(1):271–306, 1995.