A PAC RL Algorithm for Episodic POMDPs

Zhaohan Daniel Guo
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh PA 15213, USA

Shayan Doroudi
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh PA 15213, USA

Emma Brunskill
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh PA 15213, USA

Abstract

Many interesting real world domains involve reinforcement learning (RL) in partially observable environments. Efficient learning in such domains is important, but existing sample complexity bounds for partially observable RL are at least exponential in the episode length. We give, to our knowledge, the first partially observable RL algorithm with a polynomial bound on the number of episodes on which the algorithm may not achieve near-optimal performance. Our algorithm is suitable for an important class of episodic POMDPs. Our approach builds on recent advances in method of moments for latent variable model estimation.

1 INTRODUCTION

A key challenge in artificial intelligence is how to effectively learn to make a sequence of good decisions in stochastic, unknown environments. Reinforcement learning (RL) is a subfield specifically focused on how agents can learn to make good decisions given feedback in the form of a reward signal. In many important applications such as robotics, education, and healthcare, the agent cannot directly observe the state of the environment responsible for generating the reward signal, and instead only receives incomplete or noisy observations.

One important measure of an RL algorithm is its sample efficiency: how much data/experience is needed to compute a good policy and act well. One way to measure sample complexity is given by the Probably Approximately Correct framework; an RL algorithm is said to be PAC if with high probability, it selects a near-optimal action on all but a number of steps (the sample complexity) which is a polynomial function of the problem parameters. There has been substantial progress on PAC RL for the fully observable setting [Brafman and Tennenholtz, 2003, Strehl and Littman, 2005, Kakade, 2003, Strehl et al., 2012, Lattimore and Hutter, 2012], but to our knowledge there exists no published work on PAC RL algorithms for partially observable settings.

This lack of work on PAC partially observable RL is perhaps because of the additional challenge introduced by the partial observability of the environment. In fully observable settings, the world is often assumed to behave as a Markov decision process (MDP). An elegant approach for proving that a RL algorithm for MDPs is PAC is to compute finite sample error bounds on the MDP parameters. However, because the states of a partially observable MDP (POMDP) are hidden, the naive approach of directly treating the POMDP as a history-based MDP yields a state space that grows exponentially with the horizon, rather than polynomial in all POMDP parameters [Even-Dar et al., 2005].

On the other hand, there has been substantial recent interest and progress on method of moments and spectral approaches for modeling partially observable systems [Anandkumar et al., 2012, 2014, Hsu et al., 2008, Littman et al., 2001, Boots et al., 2011]. The majority of this work has focused on inference and prediction, with little work tackling the control setting. Method of moments approaches to latent variable estimation are of particular interest because for a number of models they obtain global optima and provide finite sample guarantees on the accuracy of the learned model parameters.

Inspired by the this work, we propose a POMDP RL algorithm that is, to our knowledge, the first PAC POMDP RL algorithm for episodic domains (with no restriction on the policy class). Our algorithm is applicable to a restricted but important class of POMDP settings, which include but are not limited to informa-

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

tion gathering POMDP RL domains such as preference elicitation [Boutilier, 2002], dialogue management slot-filling domains [Ko et al., 2010], and medical diagnosis before decision making [Amato and Brunskill, 2012]. Our work builds on method of moments inference techniques, but requires several non-trivial extensions to tackle the control setting. In particular, there is a subtle issue of latent state alignment: if the models for each action are learned as independent hidden Markov models (HMMs), then it is unclear how to solve the correspondence issue across latent states, which is essential for performing planning and selecting actions. Our primary contribution is to provide a theoretical analysis of our proposed algorithm, and prove that it is possible to obtain near-optimal performance on all but a number of episodes that scales as a *polynomial* function of the POMDP parameters. Similar to most fully observable PAC RL algorithms, directly instantiating our bounds would yield an impractical number of samples for a real application. Nevertheless, we believe understanding the sample complexity may help to guide the amount of data required for a task, and also similar to PAC MDP RL work, may motivate new practical algorithms that build on these ideas.

2 BACKGROUND AND RELATED WORK

The inspiration for pursuing PAC bounds for POMDPs came about from the success of PAC bounds for MDPs [Brafman and Tennenholtz, 2003, Strehl and Littman, 2005, Kakade, 2003, Strehl et al., 2012, Lattimore and Hutter, 2012]. While algorithms have been developed for POMDPs with finite sample bounds [Peshkin and Mukherjee, 2001, Even-Dar et al., 2005], unfortunately these bounds are not PAC as they have an exponential dependence on the horizon length.

Alternatively, Bayesian methods [Ross et al., 2011, Doshi-Velez, 2012] are very popular for solving POMDPs. For MDPs, there exist Bayesian methods that have PAC bounds [Kolter and Ng, 2009, Asmuth et al., 2009]; however there have been no PAC bounds for Bayesian methods for POMDPs. That said, Bayesian methods are optimal in the Bayesian sense of making the best decision given the posterior over all possible future observations, which does not translate to a frequentist finite sample bound.

We build on method of moments (MoM) work for estimating HMMs [Anandkumar et al., 2012] in order to provide a finite sample bound for POMDPs. MoM is able to obtain a global optimum, and has finite sample bounds on the accuracy of their estimates, unlike the popular Expectation-Maximization (EM) that is only guaranteed to find a local optima, and offers no finite sample guarantees. MLE approaches for estimating HMMs [Abe and Warmuth, 1992] also unfortunately do not provide accuracy guarantees on the estimated HMM parameters. As POMDP planning methods typically require us to have estimates of the underlying POMDP parameters, it would be difficult to use such MLE methods for computing a POMDP policy and providing a finite sample guarantee[1].

Aside from the MoM method in Anandkumar et al. [2012], another popular spectral method involves using Predictive State Representations (PSRs) [Littman et al., 2001, Boots et al., 2011], to directly tackle the control setting; however it only has asymptotic convergence guarantees and no finite sample analysis. There is also another method of moments approach to transfer across a set of bandits tasks, but the latent variable estimation problem is substantially simplified because the state of the system is unchanged by the selected actions [Azar et al., 2013].

Fortunately, due to the polynomial finite sample bounds from MoM, we can achieve a PAC (polynomial) sample complexity bound for POMDPs.

3 PROBLEM SETTING

We consider a partially observable Markov decision process (POMDP) which is described as the tuple (S, A, R, T, Z, b, H) where we have a set of discrete states S, discrete actions A, discrete observations Z, discrete rewards R, initial belief b (more details below), and episode length H. The transition model is represented by a set of $|A|$ matrices $T_a(i,j) : |S| \times |S|$ where the (i,j)-th entry is the probability of transitioning from s_i to s_j under action a. With a slight abuse of notation, we use Z to denote both the finite set of observations and the observation model captured by the set of $|A|$ observation matrices, Z_a where the (i,j)-th entry represents the probability of observing z_i given the agent took action a and transitioned to state s_j. We similarly do a slight abuse of notation and let R denote both the finite set of rewards, and the reward matrices R_a where the (i,j)-th entry in a matrix denotes the probability of obtaining reward r_i

[1] Abe and Warmuth [1992]'s MLE approach guarantees that the estimated probability over H-length observation sequences has a bounded KL-divergence from the true probability of the sequence under the true parameters, which is expressed as a function of the number of underlying data samples used to estimate the HMM parameters. We think it may be possible to use such estimates in the control setting when modeling hidden state control systems as PSRs, and employing a forward search approach to planning; however, there remain a number of subtle issues to address to ensure such an approach is viable and we leave this as an interesting direction for future work.

when taking action a in state s_j. Note that in our setting we also treat the reward as an additional observation[2].

The objective in POMDP planning is to compute a policy π that achieves a large expected sum of future rewards, where π is a mapping from histories of prior sequences of actions, observations, and rewards, to actions. In many cases we capture prior histories using a sufficient statistic called the belief b where $b(s)$ represents the probability of being in a particular state s given the prior history of actions, observations and rewards. One popular method for POMDP planning involves representing the value function by a finite set of α-vectors, where $\alpha(s)$ represents the expected sum of future rewards of following the policy associated with the α-vector from initial state s. POMDP planning then proceeds by taking the first action associated with the policy of the α-vector which yields the maximum expected value for the current belief state, which can be computed for a particular α-vector using the dot product $\langle b, \alpha \rangle$.

In the reinforcement learning setting, the transition, observation, and/or reward model parameters are initially unknown. The goal is to learn a policy that achieves large sum of rewards in the environment without advance knowledge of how the world works.

We make the following assumptions about the domain and problem setting:

1. We consider episodic, finite horizon partially observable RL (PORL) settings
2. It is possible to achieve a non-zero probability of being in any state in two steps from the initial belief.
3. For each action a, the transition matrix T_a is full rank, and the observation matrix Z_a and reward matrix R_a are full column rank.

The first assumption on the setting is satisfied by many real world situations involving an agent repeatedly doing a task: for example, an agent may sequentially interact with many different customers each for a finite amount of time. The key restrictions on the setting are captured in assumptions 2 and 3. Assumption 2 is similar to a mixing assumption and is necessary in order for MoM to estimate dynamics for all states. Assumption 3 is necessary for MoM to uniquely determine the transition, observation, and reward dynamics. The second assumption may sound quite strong, as in some POMDP settings states are only reachable by a complex sequence of carefully chosen actions, such as in robotic navigation or video games. However, assumption 2 is commonly satisfied in many important POMDP settings that primarily involve information gathering. For example, in preference elicitation or user modeling, POMDPs are commonly used to identify the, typically static, hidden intent or preference or state of the user, before taking some action based on the resulting information [Boutilier, 2002]. Examples of this include dialog systems [Ko et al., 2010], medical diagnosis and decision support [Amato and Brunskill, 2012], and even human-robot collaboration preference modeling [Nikolaidis et al., 2015]. In such settings, the belief commonly starts out non-zero over all possible user states, and slowly gets narrowed down over time. The third assumption is also significant, but is still satisfied by an important class of problems that overlap with the settings captured by assumption 2. Information gathering POMDPs where the state is hidden but static automatically satisfy the full rank assumption on the transition model, since it is an identity matrix. Assumption 3 on the observation and reward matrices imply that the cardinality of the set of observations (and rewards) is at least as large as the size of the state space. A similar assumption has been made in many latent variable estimation settings (e.g. [Anandkumar et al., 2012, 2014, Song et al., 2010]) including in the control setting [Boots et al., 2011]. Indeed, when the observations consist of videos, images or audio signals, this assumption is typically satisfied [Boots et al., 2011], and such signals are very common in dialog systems and the user intent and modeling situations covered by assumption 2. Satisfying that the reward matrix has full rank is typically trivial as the reward signal is often obtained by discretizing a real-valued reward. Therefore, while we readily acknowledge that our setting does not cover all generic POMDP reinforcement learning settings, we believe it does cover an important class of problems that are relevant to real applications.

4 ALGORITHM

Our goal is to create an algorithm that can achieve near optimal performance from the initial belief on each episode. Prior work has shown that the error in the POMDP value function is bounded when using model parameter estimates that themselves have bounded error [Ross et al., 2009, Fard et al., 2008];

[2]In planning problems the reward is typically a real-valued scalar, but in PORL we must learn the reward model. This requires assuming some mapping between states and rewards. For simplicity we assume multinomial distribution over a discrete set of rewards. Note that we can always discretized a real-valued reward into a finite set of values with bounded error on the resulting value function estimates, and our choice makes very little restrictions on the underlying setting.

Algorithm 1: EEPORL

input: $S, A, Z, R, H, N, c, \pi_{rest}$

Let $\pi_{explore}$ be the policy where a_1, a_2 are uniformly random, and $p(a_{t+2}|a_t) = \frac{1}{1+c|A|}(\mathbf{I} + c\mathbf{1}_{|A|\times|A|})$;

$X \leftarrow \emptyset$;

`// Phase 1:`

for *episode* $i \leftarrow 1$ **to** N **do**

  Follow $\pi_{explore}$ for 4 steps ;

  Let $x_t = (a_t, r_t, z_t, a_{t+1})$;

  $X \leftarrow X \cup \{(x_1, x_2, x_3)\}$;

  Execute π_{rest} for the rest of the steps ;

`// Phase 2:`

Get $\widehat{T}, \widehat{O}, \widehat{w}$ for the induced HMM from X through our extended MoM method ;

Using the labeling from Algorithm 2 with $\widehat{O}$, compute estimated POMDP parameters.;

Call Algorithm 3 with estimated POMDP parameters to estimate a near optimal policy $\widehat{\pi}$;

Execute $\widehat{\pi}$ for the rest of the episodes ;

Algorithm 2: LabelActions

input: $\widehat{O}$

foreach *column* i *of* $\widehat{O}$ **do**

  Find a row j such that $\widehat{O}(i,j) \geq \frac{2}{3|R||Z|}$;

  Let the observation associated with row j be (a, r', z', a'), label column i with (a, a') ;

Algorithm 3: FindPolicy

input: $\widehat{b}(s_{(a_0,a_1)})$, $\widehat{p}(z|a, s_{(a,a')})$, $\widehat{p}(r|s_{(a,a')}, a')$, $\widehat{p}(s_{(a',a'')}|s_{(a,a')}, a')$

$\forall a_-, a \in A, \quad \Gamma_1^{a_-,a} = \{\widehat{\beta}_1^a(s_{(a_-,a)})\}$;

for $t \leftarrow 2$ **to** H **do**

  $\forall a, a' \in A, \Gamma_t^{a,a'} = \emptyset$;

  for $a, a' \in A$ **do**

    for $f_t(r,z) \in (|R| \times |Z| \to \Gamma_{t-1}^{a,a'})$ **do**

      `// all mappings from an observation pair to a previous` β`-vector`

      $\forall a_- \in A, \Gamma_t^{a_-,a} = \Gamma_t^{a_-,a} \cup \{\beta_t^{a,f_t}(s_{(a_-,a)})\}$;

Return $\arg\max_{a_0,a_1,\beta_H(s_{(a_0,a_1)}))\in\Gamma_H^{a_0,a_1}} (\widehat{b} \cdot \beta_H)$;

however, this work takes a sensitivity analysis perspective, and does not address how such model estimation errors themselves could be computed or bounded.[3]

[3] Fard et al. [2008] assume that labels of the hidden states are provided, which removes the need for latent variable estimation.

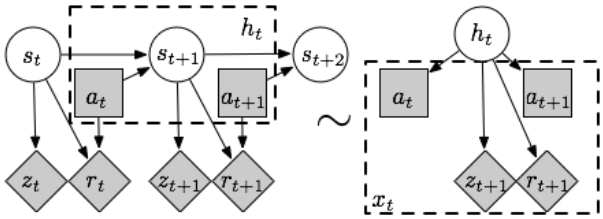

Figure 1: POMDP (left) analogous to induced HMM (right). Gray nodes show fully observed variables, whereas white nodes show latent states.

In contrast, many PAC RL algorithms for MDPs have shown that exploration is critical in order to get enough data to estimate the model parameters. However in MDPs, algorithms can directly observe how many times every action has been tried in every state, and can use this information to steer exploration towards less explored areas. In partially observable settings it is more challenging, as the state itself is hidden, and so it is not possible to directly observe the number of times an action has been tried in a latent state. Fortunately, recent advances in method of moments (MoM) estimation procedures for latent variable estimation (see e.g. [Anandkumar et al., 2012, 2014]) have demonstrated that in certain uncontrolled settings, including many types of hidden Markov models (HMMs), it is still possible to achieve accuracy estimates of the underlying latent variable model parameters as a function of the amount of data samples used to perform the estimation. For some intuition about this, consider starting in a belief state b which has non-zero probability over all possible states. If one can repeatedly take the same action a from same belief b, given a sufficient number of samples, we will have actually taken action a in each state many times (even if we don't know the specific instances on which action a was taken in a state s).

The control setting is more subtle than the uncontrolled setting which has been the focus of the majority of recent MoM spectral learning research, because we wish to estimate not just the transition and observation models of a HMM, but to estimate the POMDP model parameters. Our ultimate interest is in being able to select good actions. A naive approach is to independently learn the transition, observation, and reward parameters for each separate action, by restricting the POMDP to only execute a single action, thereby turning the POMDP into an HMM. However, this simple aproach fails because the returned parameters can correspond to a different labeling of the hidden states. For example, the first column of the transition matrix for action a_1 may actually correspond to the state s_2, while the first column of the transition ma-

trix for action a_2 may truly correspond to s_5. We require that the labeling must be consistent for all actions since we wish to compute what happens when different actions are executed consecutively. An unsatisfactory way to match up the labels for different actions is by requiring that the initial belief state have probabilities that are unique and well separated per state. Then we can use the estimated initial belief from each action to match up the labels. However, this is a very strong assumption on the starting belief state which is unlikely to be realized.

To address this challenge of mismatched labels, we transform our POMDP into an induced HMM (see Figure 1) by fixing the policy to $\pi_{explore}$ (for a few steps, during a certain number of episodes), and create an alternate hidden state representation that directly solves the problem of alignment of hidden states across actions. Specifically, we make the hidden state at time t of the induced HMM, denoted by h_t, equal to the tuple of the action at time step t, the next state, and the subsequent action, $h_t = (a_t, s_{t+1}, a_{t+1})$. We denote the observations of the induced HMM by x, and the observation associated with a hidden state h_t is the tuple $x_t = (a_t, r_t, z_t, a_{t+1})$. Figure 1 shows how the graphical model of our original POMDP is related to the graphical model of the induced HMM. In making this transformation, our resulting HMM still satisfies the Markov assumption: the next state is only a function of the prior state, and the observation is only a function of the current state. But, this transformation also has the desired property that it is now possible to directly align the identity of states across selected actions. This is because HMM parameters now depend on both state and action, so there is a built-in correlation between different actions. We will discuss this more in the theoretical analysis.

We are now ready to describe our algorithm for episodic finite horizon reinforcement learning in POMDPs, EEPORL (Explore then Exploit Partially Observable RL, which is shown in Algorithm 1). Our algorithm is model-based and proceeds in two phases. In the first phase, it performs exploration to collect samples of trying different actions in different (latent) states. After the first phase completes, we extend a MoM approach [Anandkumar et al., 2012] to compute estimates of the induced HMM parameters. We use these estimates to obtain a near-optimal policy.

4.1 Phase 1

The first phase consists of the first N episodes. Let $\pi_{explore}$ be a fixed open-loop policy for the first four actions of an episode. In $\pi_{explore}$ actions a_1, a_2 are selected uniformly at random, and $p(a_{t+2}|a_t) = \frac{1}{1+c|A|}(\mathbf{I} + c\mathbf{1}_{|A|\times|A|})$ where c can be any positive real number. For our proof, we pick $c = O(1/|A|)$. Note that $\pi_{explore}$ only depends on previous actions and not on any observations. The definition of $p(a_{t+2}|a_t)$ for what will work for the proof only requires it to be full-rank and having some minimum probability over all actions. We chose a perturbed identity matrix for simplicity. Since $\pi_{explore}$ is a fixed policy, the POMDP process reduces to a HMM for these first four steps. During these steps we store the observed experience as (x_1, x_2, x_3), where $x_t = (a_t, r_t, z_t, a_{t+1})$ is an observation of our previously defined induced HMM. The algorithm then follows policy π_{rest} for the remaining steps of the episode. All of these episodes will be considered as potentially non-optimal, and so the choice of π_{rest} does not impact the theoretical analysis. However, empirically π_{rest} could be constructed to encourage near optimal behavior given the observed data collected up to the current episode.

4.2 Parameter Estimation

After Phase 1 completes, we have N samples of the tuple (x_1, x_2, x_3). We then apply our extension to the MoM algorithm for HMM parameter estimation by Anandkumar et al. [2012]. Our extension computes estimates and bounds on the transition model $\widehat{T}$ which is not computed in the original method. To summarize, this procedure yields an estimated transition matrix $\widehat{T}$, observation matrix $\widehat{O}$, and belief vector $\widehat{w}$ for the induced HMM. The belief $\widehat{w}$ is over the second hidden state, h_2.

As mentioned before as one major challenge, labeling of the states h of the induced HMM is arbitrary; however it is consistent between $\widehat{T}, \widehat{O}, \widehat{w}$ since this is a single HMM inference problem. Recall that a hidden state in our induced HMM is defined as $h_t = (a_t, s_{t+1}, a_{t+1})$. Since the actions are fully observable, it is possible to label each state $h = (a, s', a')$ (i.e. the columns of $\widehat{O}$, the rows and columns of $\widehat{T}$, and the rows of $\widehat{w}$) with two actions (a, a') that are associated with that state. This is possible because the true observation matrix entries for the actions of a hidden state must be non-zero, and the true value of all other entries (for other actions) must be zero; therefore, as long as we have sufficiently accurate estimates of the observation matrix, we can use the observation matrix parameters to augment the states h with their associated action pair. This procedure is performed by Algorithm 2. This labeling provides a connection between the HMM state h and the original POMDP state. For a particular pair of actions a, a', there are exactly $|S|$ HMM states that correspond to them. Thus looking at the columns of $\widehat{O}$ from left-to-right, and only picking out the columns that are labeled with a, a' results in a

specific ordering of the states $(a, \cdot, a')$, which is a permutation of the POMDP states, which we denote as $\{s_{(a,a'),1}, s_{(a,a'),2}, \ldots, s_{(a,a'),|S|}\}$. We will also use the notation $s_{(a,a')}$ to implicitly refer to a vector of states in the order of the permutation.

The algorithm proceeds to estimate the original POMDP parameters in order to perform planning and compute a policy. Note that the estimated parameters use the computed $s_{(a,a')}$ permutations of the state. Let $\widehat{O}^{a,a'}$ be the submatrix where the rows and columns correspond to the actions (a, a') and $\widehat{T}^{a,a',a''}$ be the submatrix where the rows correspond to the actions (a', a'') and columns correspond to the actions (a, a'). Then the estimated POMDP parameters can be computed as follows:

$$\widehat{b}(s_{(a_0,a_1)}) = normalize((\widehat{T}^{-1}\widehat{T}^{-1}\widehat{w})(a_0, \cdot, a_1))$$

$$\widehat{p}(z|a, s_{(a,a')}) = normalize(\sum_r \widehat{O}^{a,a'})$$

$$\widehat{p}(r|s_{(a,a')}, a') = normalize(\sum_z \widehat{O}^{a,a'})$$

$$\widehat{p}(s_{(a',a'')}|s_{(a,a')}, a') = normalize(\widehat{T}^{a,a',a''})$$

Note that we require an additional *normalize*() procedure since the MoM approach we leverage is not guaranteed to return well formed probability distributions. The normalization procedure just divides by the sum to make them into valid probability distributions (if there are negative values we can either set them to zero or even just use the absolute value).

Algorithm 3 then uses these estimated POMDP parameters to compute a policy. The algorithm constructs β-vectors (see Definition 1) that represent the expected sum of rewards of following a particular policy starting with action a' given an input permuted state $s_{(a,a')}$. Aside from this slight modification, β-vectors are analogous to α-vectors in standard POMDP planning. The β-vectors form an approximate value function for the underlying POMDP and can be used in a similar way to standard α-vectors.

4.3 Phase 2

In phase 2, after estimating the POMDP parameters and β-vectors, we use the estimated POMDP value function to extract a policy for acting, and we will shortly prove sufficient conditions for this policy to be near-optimal for all remaining episodes.

The policy followed depends on the computed value function. If computationally tractable, one can compute β-vectors incrementally for all possible H-step policies. In this case, control proceeds by finding the best β-vector for the estimated initial belief $\widehat{b}(s_{(a_0,a_1)})$ (largest dot product of the β-vector with the initial belief) and then following the associated policy $\widehat{\pi}$. $\widehat{\pi}$ is then followed for the entire episode with no additional belief updating required as the policy itself encodes the conditional branching.

However, in practical circumstances, it will not be possible to enumerate all possible H-step policies. In this case, one can use point-based approaches or other methods that use α-vectors to enumerate only a subset of possible policies. In this case there will be an additional error $\epsilon_{planning}$ in the final error bound due to finite set of policies considered. In our analysis we omit $\epsilon_{planning}$ for simplicity and assume that we enumerate all H-step policies.

Definition 1. *A β-vector taking as input $s_{(a,a')}$ with root action a' and t-step conditional policies $f_t(r, z)$ for each observation pair (r, z) is defined as*

$$\beta_1^{a'}(s_{(a,a')}) = \sum_r p(r|s_{(a,a')}, a) \cdot r$$

$$\beta_{t+1}^{a',f_t}(s_{(a,a')}) = \sum_{r,z,s_{(a',f_t(r,z))}} (r + \gamma\beta_t^{f_t(r,z)}(s_{(a',f_t(r,z))})) \cdot p(r|s_{(a,a')}, a)p(z|s_{(a',f_t(r,z))}, a)p(s_{(a',f_t(r,z))}|s_{(a,a')}, a)$$

where $f_t(r, z)$ can also denote the root action of the policy $f_t(r, z)$ used in terms like $s_{(a,f_t(r,z))}$.

5 THEORY

5.1 PAC Theorem Setup

We now state our primary result. For full details, please refer to our tech report[4]. Before doing so, we define some additional notation. Let $V^\pi(b) = \sum_{i=1}^H r_t$ starting from belief b be the total undiscounted reward following policy π for an episode. Let $\sigma_{1,a}(T_a) = \max_a \sigma_1(T_a)$ and similarly for $\sigma_{1,a}(R_a)$ and $\sigma_{1,a}(Z_a)$. Let $\underline{\sigma}_a(T_a) = \min_a \sigma_{|S|}(T_a)$ and similarly for $\underline{\sigma}_a(R_a)$ and $\underline{\sigma}_a(Z_a)$. Assume $\underline{\sigma}_a(T_a)$, $\underline{\sigma}_a(R_a)$, and $\underline{\sigma}_a(Z_a)$ are all at most 1 (otherwise each term can be replaced by 1 in the final sample complexity bound below).

5.2 PAC Theorem

Theorem 1. *For POMDPs that satisfy the stated assumptions defined in the problem setting, executing EEPORL will achieve an expected episodic reward of $V(b_0) \geq V^*(b_0) - \epsilon$ on all but a number of episodes that is bounded by*

$$O\left(\frac{H^4 V_{\max}^2 |A|^{12}|R|^4|Z|^4|S|^{12}\left(1 + \sqrt{\log(\frac{3}{\delta})}\right)^2 \log\left(\frac{3}{\delta}\right)}{C_{d,d,d}\left(\frac{\delta}{3}^2\right)\underline{\sigma}_a(T_a)^6\underline{\sigma}_a(R_a)^8\underline{\sigma}_a(Z_a)^8\epsilon^2}\right)$$

[4] `http://www.cs.cmu.edu/~zguo/#publications`

with probability at least $1-\delta$*, where*

$$C_{d,d,d}(\delta) = \min(C_{1,2,3}(\delta), C_{1,3,2}(\delta))$$
$$C_{1,2,3}(\delta) = \min\left(\frac{\min_{i\neq j}||M_3(\vec{e}_i - \vec{e}_j)||_2 \cdot \sigma_k(P_{1,2})^2}{||P_{1,2,3}||_2 \cdot k^5 \cdot \kappa(M_1)^4} \cdot \frac{\delta}{\log(k/\delta)}, \frac{\sigma_k(P_{1,3})}{1}\right)$$
$$C_{1,3,2}(\delta) = \min\left(\frac{\min_{i\neq j}||M_2(\vec{e}_i - \vec{e}_j)||_2 \cdot \sigma_k(P_{1,3})^2}{||P_{1,3,2}||_2 \cdot k^5 \cdot \kappa(M_1)^4} \cdot \frac{\delta}{\log(k/\delta)}, \frac{\sigma_k(P_{1,2})}{1}\right)$$

The quantities $C_{1,2,3}, C_{1,3,2}$ directly arise from using the previously referenced MoM method for HMM parameter estimation [Anandkumar et al., 2012] and involve singular values of the moments of the induced HMM and the induced HMM parameters (see [Anandkumar et al., 2012] for details).

We now briefly overview the proof. Detailed proofs are available in the supplemental material. We first show that by executing EEPORL we obtain parameter estimates of the induced HMM, and bounds on these estimates, as a function of the number of data points (Lemma 2). We then prove that we can use the induced HMM to obtain estimated parameters of the underlying POMDP (Lemma 4). Then we show that we can compute policies that are equivalent (in structure and value) to those from the original POMDP (Lemma 5). We then bound the error in the resulting value function estimates of the resulting policies due to the use of approximate (instead of exact) model parameters (Lemma 6). This allows us to compute a bound on the number of required samples (episodes) necessary to achieve near-optimal policies, with high probability, for use in phase 2.

We commence the proof by bounding the error in estimates of the induced HMM parameters. In order to do that, we introduce Lemma 1, which proves that samples taken in phase 1 belong to an induced HMM where the transition and observation matrices are full rank. This is a requirement for being able to apply the MoM HMM parameter estimation procedure of Anandkumar et al. [2012].

Lemma 1. *The induced HMM has the observation and transition matrices defined as*

$$O(x_t^i, h_t^j) = \delta(a_t^i, a_t^j)\delta(a_{t+1}^i, a_{t+1}^j)p(z_{t+1}^i|a_t^j, s_{t+1}^j)p(r_{t+1}^i|s_{t+1}^j, a_{t+1}^j)$$
$$T(h_{t+1}^i, h_t^j) = \delta(a_{t+1}^i, a_{t+1}^j)p(s_{t+2}^i|s_{t+1}^j, a_{t+1}^j)p(a_{t+2}^i|a_t^j)$$

where i *is the index over the rows and* j *is the index over the columns, and* $x_t^i = (a_t^i, z_{t+1}^i, r_{t+1}^i, a_{t+1}^i)$, $h_{t+1}^i = (a_{t+1}^i, s_{t+2}^i, a_{t+2}^i)$, $h_t^j = (a_t^j, s_{t+1}^j, a_{t+1}^j)$. T *and* O *are both full rank and* $w = p(h_2)$ *has positive probability everywhere. Furthermore the following terms are bounded:* $\|T\|_2 \leq \sqrt{|S|}$, $\|T^{-1}\|_2 \leq \frac{2(1+c|A|)}{\underline{\sigma}_a(T_a)}$, $\sigma_{\min}(O) \geq \underline{\sigma}_a(R_a)\underline{\sigma}_a(Z_a)$, *and* $\|O\|_2 = \sigma_1(O) \leq |S|$.

Next, we use Lemma 2, which is an extension of the method of moments method by Anandkumar et al. [2012] that provides a bound on the accuracies of the estimated induced HMM parameters in terms of N, the number of samples collected. Our extension involves computing $\widehat{T}$ (the original method only had $\widehat{O}$ and $\widehat{OT}$) and bounding its accuracy.

Lemma 2. *Given an HMM such that* $p(h_2)$ *has positive probability everywhere, the transition matrix is full rank, and the observation matrix is full column rank, then by gathering* N *samples of* (x_1, x_2, x_3)*, the estimates* $\widehat{T}, \widehat{O}, \widehat{w}$ *can be computed such that*

$$||\widehat{T} - T||_2 \leq 18|A||S|^4(\underline{\sigma}_a(R_a)\underline{\sigma}_a(Z_a))^{-4}\epsilon_1$$
$$||\widehat{O} - O||_2 \leq |A||S|^{0.5}\epsilon_1$$
$$||\widehat{O} - O||_{\max} \leq \epsilon_1$$
$$||\widehat{w} - w||_2 \leq 14|A|^2|S|^{2.5}(\underline{\sigma}_a(R_a)\underline{\sigma}_a(Z_a))^{-4}\epsilon_1$$

where $||\cdot||_2$ *is the spectral norm for matrices, and the euclidean norm for vectors, and* w *is the marginal probability of* h_2*, with probability* $1-\delta$*, as long as*

$$N \geq O\left(\frac{|A|^2|Z||R|(1+\sqrt{\log(1/\delta)})^2}{(C_{d,d,d}(\delta))^2 \cdot \epsilon_1^2}\log\left(\frac{1}{\delta}\right)\right)$$

Next we proceed by showing how to bound the error in the estimates of the POMDP parameters. The following Lemma 3 is a prerequisite for computing the submatrices of $\widehat{O}$ and $\widehat{T}$ needed for the estimates of the POMDP parameters.

Lemma 3. *Given* $\widehat{O}$ *with max-norm error* $\epsilon_O \leq \frac{1}{3|Z||R|}$*, then the columns which correspond to HMM states of the form* $h = (a, s', a')$ *can be labeled with their corresponding* a, a' *using Algorithm 2.*

With the correct labels, the submatrices of $\widehat{O}$ and $\widehat{T}$ allow us to compute estimates of the original POMDP parameters in terms of these permutations $s_{(a,a')}$. Lemma 4 bounds the error in these resulting estimates.

Lemma 4. *Given* $\widehat{T}, \widehat{O}, \widehat{w}$ *with max-norm errors* $\epsilon_T, \epsilon_O, \epsilon_w$ *respectively, then the following bounds hold on the estimated POMDP model parameters with prob-*

ability at least $1-\delta$

$$|\widehat{p}(s_{(a',a'')}|s_{(a,a')},a') - p(s_{(a',a'')}|s_{(a,a')},a')| \leq \frac{4|S|\epsilon_T}{\epsilon_a^2}$$
$$|\widehat{p}(z|a,s_{(a,a')}) - p(z|a,s_{(a,a')})| \leq 4|Z||R|\epsilon_O$$
$$|\widehat{p}(r|s_{(a,a')},a') - p(r|s_{(a,a')},a')| \leq 4|Z||R|\epsilon_O$$
$$|\widehat{b}(s_{(a_0,a_1)}) - b(s_{(a_0,a_1)})| \leq 4|A|^4|S|(||T^{-1}||_2^2\epsilon_w + 6||T^{-1}||_2^3\epsilon_T)$$

where $\epsilon_a = \Theta(1/|A|)$

We proceed by bounding the error in computing the estimated β-vectors. Lemma 5 states that β-vectors are equivalent under permutation to α-vectors.

Lemma 5. *Given the permutation of the states* $s_{(a,a'),j} = s_{\phi((a,a'),j)}$*,* β*-vectors and* α*-vectors over the same policy* π_t *are equivalent i.e.* $\beta_t^{\pi_t}(s_{(a,a'),j}) = \alpha_t^{\pi_t}(s_{\phi((a,a'),j)})$

The following lemma bounds the error in the resulting α-vectors obtained by performing POMDP planning, and follows from prior work [Fard et al., 2008, Ross et al., 2009].

Lemma 6. *Suppose we have approximate POMDP parameters with errors* $|\widehat{p}(s'|s,a) - p(s'|s,a)| \leq \epsilon_T$*,* $|\widehat{p}(z|a,s') - p(z|a,s')| \leq \epsilon_Z$*, and* $|\widehat{p}(r|s,a) - p(r|s,a')| \leq \epsilon_R$*. Then for any* t*-step conditional policy* π_t

$$|\alpha_t^{\pi_t}(s) - \widehat{\alpha}_t^{\pi_t}(s)| \quad \leq t^2 R_{\max}(|R|\epsilon_R + |S|\epsilon_T + |Z|\epsilon_Z).$$

We next prove that our EEPORL algorithm computes a policy that is optimal for the input parameters[5]:

Lemma 7. *Algorithm 3 finds the policy* $\widehat{\pi}$ *which maximizes* $V^{\widehat{\pi}}(\widehat{b}(s_1))$ *for a POMDP with parameters* $\widehat{b}(s_1), \widehat{p}(z|a,s'), \widehat{p}(r|s,a)$*, and* $\widehat{p}(s'|s,a)$*.*

We now have all the key pieces to prove our result.

Proof. (Proof sketch of Theorem 1). Lemma 4 shows that the error in the estimates of the POMDP parameters can be bounded in terms of the error in the induced HMM parameters, which is itself bounded in terms of the number of samples (Lemma 1). Lemma 5 and Lemma 6 together bound in the error in computing the estimated value function (as represented by β-vectors) using estimated POMDP parameters.

We then need to bound the error from executing $\widehat{\pi}$ that Algorithm 3 returns compared to the optimal policy π^*. We know from Lemma 7 that Algorithm 3 correctly identifies the best policy for the estimated POMDP. Then let the initial beliefs $b, \widehat{b}$ have error $\|b - \widehat{b}\|_\infty \leq \epsilon_b$, and the bound over α-vectors of any policy π, $\|\alpha^\pi - \widehat{\alpha}^\pi\|_\infty \leq \epsilon_\alpha$ be given. Then

$$\begin{aligned}
\widehat{V}^{\widehat{\pi}}(\widehat{b}) &= \widehat{b}\cdot\widehat{\alpha}^{\widehat{\pi}} \geq \widehat{b}\cdot\widehat{\alpha}^{\pi^*} \\
&\geq \widehat{b}\cdot\alpha^{\pi^*} - |\widehat{b}\cdot\alpha^{\pi^*} - \widehat{b}\cdot\widehat{\alpha}^{\pi^*}| \geq \widehat{b}\cdot\alpha^{\pi^*} - \epsilon_\alpha \\
&\geq b\cdot\alpha^{\pi^*} - |b\cdot\alpha^{\pi^*} - \widehat{b}\cdot\alpha^{\pi^*}| - \epsilon_\alpha \\
&\geq b\cdot\alpha^{\pi^*} - \epsilon_b V_{\max} - \epsilon_\alpha = V^*(b) - \epsilon_b V_{\max} - \epsilon_\alpha
\end{aligned}$$

where the first inequality is because $\widehat{\pi}$ is the optimal policy for $\widehat{b}$ and $\widehat{\alpha}$, the second inequality is by the triangle inequality, the third inequality is because $\|\widehat{b}\|_1 = 1$, the fourth inequality is by the triangle inequality, the fifth inequality is since α is at most $V_{\max}$. Next

$$\begin{aligned}
V^{\widehat{\pi}}(b) = b\cdot\alpha^{\widehat{\pi}} &\geq \widehat{b}\cdot\alpha^{\widehat{\pi}} - |\widehat{b}\cdot\alpha^{\widehat{\pi}} - b\cdot\alpha^{\widehat{\pi}}| \\
&\geq \widehat{b}\cdot\alpha^{\widehat{\pi}} - \epsilon_b V_{\max} \\
&\geq \widehat{b}\cdot\widehat{\alpha}^{\widehat{\pi}} - |\widehat{b}\cdot\widehat{\alpha}^{\widehat{\pi}} - \widehat{b}\cdot\alpha^{\widehat{\pi}}| - \epsilon_b V_{\max} \\
&\geq \widehat{b}\cdot\widehat{\alpha}^{\widehat{\pi}} - \epsilon_\alpha - \epsilon_b V_{\max}
\end{aligned}$$

where the first inequality is by triangle inequality, the second inequality is because α is at most $V_{\max}$, the third inequality is triangle inequality, and the fourth inequality is due to $\|\widehat{b}\|_1 = 1$. Putting those two together results in

$$V^{\widehat{\pi}}(b) \quad \geq V^*(b) - 2\epsilon_b V_{\max} - 2\epsilon_\alpha$$

Letting $\epsilon = 2\epsilon_b V_{\max} + 2\epsilon_\alpha$, and setting the number of episodes N to the value specified in the theorem will ensure that the resulting errors ϵ_b and ϵ_α are small enough to obtain an ϵ-optimal policy as desired. □

6 CONCLUSION

We have provided a PAC RL algorithm for an important class of episodic POMDPs, which includes many information gathering domains. To our knowledge this is the first RL algorithm for partially observable settings that has a sample complexity that is a polynomial function of the POMDP parameters.

There are many areas for future work. We are interested in reducing the set of currently required assumptions, thereby creating PAC PORL algorithms that are suitable to more generic settings. Such a direction may also require exploring alternatives to method of moments approaches for performing latent variable estimation. We also hope that our theoretical results will lead to further insights on practical algorithms for partially observable RL.

[5]Again, we could easily modify this to account for approximate planning error, but leave this out for simplicity, as we do not expect this to make a significant impact on the resulting sample complexity, except in terms of minor changes to the polynomial terms.

References

Naoki Abe and Manfred K Warmuth. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 9(2-3):205–260, 1992.

Christopher Amato and Emma Brunskill. Diagnose and decide: An optimal bayesian approach. In *In Proceedings of the Workshop on Bayesian Optimization and Decision Making at the Twenty-Sixth Annual Conference on Neural Information Processing Systems (NIPS-12)*, 2012.

Animashree Anandkumar, Daniel Hsu, and Sham M Kakade. A method of moments for mixture models and hidden markov models. *arXiv preprint arXiv:1203.0683*, 2012.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

John Asmuth, Lihong Li, Michael L Littman, Ali Nouri, and David Wingate. A bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 19–26. AUAI Press, 2009.

Mohammad Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2220–2228. Curran Associates, Inc., 2013.

Byron Boots, Sajid M Siddiqi, and Geoffrey J Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

Craig Boutilier. A pomdp formulation of preference elicitation problems. In *AAAI/IAAI*, pages 239–246, 2002.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

Finale Doshi-Velez. *Bayesian nonparametric approaches for reinforcement learning in partially observable domains*. PhD thesis, Massachusetts Institute of Technology, 2012.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Reinforcement learning in pomdps without resets. In *IJCAI*, pages 690–695, 2005.

Mahdi Milani Fard, Joelle Pineau, and Peng Sun. A variance analysis for pomdp policy evaluation. In *AAAI*, pages 1056–1061, 2008.

Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *arXiv preprint arXiv:0811.4413*, 2008.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning.* . PhD thesis, University College London, 2003.

Li Ling Ko, David Hsu, Wee Sun Lee, and Sylvie CW Ong. Structured parameter elicitation. In *AAAI*, 2010.

J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 513–520. ACM, 2009.

Tor Lattimore and Marcus Hutter. Pac bounds for discounted mdps. In *Algorithmic learning theory*, pages 320–334. Springer, 2012.

Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *NIPS*, volume 14, pages 1555–1561, 2001.

Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *HRI*, pages 189–196, 2015.

Leonid Peshkin and Sayan Mukherjee. Bounds on sample size for policy evaluation in markov environments. In *Computational Learning Theory*, pages 616–629. Springer, 2001.

Stephane Ross, Masoumeh Izadi, Mark Mercer, and David Buckeridge. Sensitivity analysis of pomdp value functions. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, pages 317–323. IEEE, 2009.

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.

L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert space embeddings of hidden Markov models. In *Proc. 27th Intl. Conf. on Machine Learning (ICML)*, 2010.

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

Alexander L Strehl, Lihong Li, and Michael L Littman. Incremental model-based learners with formal learning-time guarantees. *arXiv preprint arXiv:1206.6870*, 2012.