
Experimental results: Reinforcement Learning of POMDPs using Spectral Methods

Kamyar Azizzadenesheli
University of California, Irvine

Alessandro Lazaric
INRIA, France

Animashree Anandkumar
University of California, Irvine

Abstract

We propose a new reinforcement learning algorithm for partially observable Markov decision processes (POMDP) based on spectral decomposition methods. While spectral methods have been previously employed for consistent learning of (passive) latent variable models such as hidden Markov models, POMDPs are more challenging since the learner interacts with the environment and possibly changes the future observations in the process. We devise a learning algorithm running through epochs, in each epoch we employ spectral techniques to learn the POMDP parameters from a trajectory generated by a fixed policy. At the end of the epoch, an optimization oracle returns the optimal memoryless planning policy which maximizes the expected reward based on the estimated POMDP model. We prove an order-optimal regret bound with respect to the optimal memoryless policy and efficient scaling with respect to the dimensionality of observation and action spaces.

1 Introduction

Reinforcement Learning (RL) is an effective approach to solve the problem of sequential decision-making under uncertainty. RL agents learn how to maximize long-term reward using the experience obtained by direct interaction with a stochastic environment (Sutton and Barto, 1998). Since the environment is initially unknown, the agent has to balance between *exploring* the environment to estimate its structure, and *exploiting* the estimates to compute a policy that maximizes the long-term reward. As a result, designing a RL algorithm requires three different elements: **1)** an estimator for the environment’s structure, **2)** a planning algorithm to compute the optimal policy of the estimated environment (LaValle, 2006), and **3)** a strategy to make a trade off between exploration and exploitation to minimize the *regret*, i.e., the difference between the performance of the exact optimal policy and the rewards accumulated by the agent over time.

Most of RL literature assumes that the environment can be modeled as a Markov decision process (MDP), with a Markovian state evolution that is fully observed. A number of exploration-exploitation strategies have been shown to have strong performance guarantees for MDPs, either in terms of regret or sample complexity (see Sect. 1.2 for a review). However, the assumption of full observability of the state evolution is often violated in practice, and the agent may only have noisy observations of the true state of the environment (e.g., noisy sensors in robotics). In this case, it is more appropriate to use the partially-observable MDP or POMDP (Sondik, 1971) model.

Many challenges arise in designing RL algorithms for POMDPs. Unlike in MDPs, the estimation problem (element 1) involves identifying the parameters of a latent variable model (LVM). In an MDP the agent directly observes (stochastic) state transitions, and the estimation of the generative model is straightforward via empirical estimators. On the other hand, in a POMDP the transition and reward models must be inferred from noisy observations and the Markovian state evolution is hidden. The planning problem (element 2), i.e., computing the optimal policy for a POMDP with known parameters, is PSPACE-complete (Papadimitriou and Tsitsiklis, 1987), and it requires

solving an augmented MDP built on a continuous belief space (i.e., a distribution over the hidden state of the POMDP). Finally, integrating estimation and planning in an exploration–exploitation strategy (element 3) with guarantees is non-trivial and no no-regret strategies are currently known (see Sect. 1.2). To handle these challenges, we build up the results in this paper on the top of the previous paper Azzadenesheli et al. (2016b) on RL of POMDPs.

1.1 Summary of Results

The main contributions of this paper are as follows: We propose a new RL algorithm for POMDPs that incorporates spectral parameter estimation within a exploration-exploitation framework. Then we apply this algorithm on a grid world atari game and compare its performance with state of the art Deep Q Learning (DQN) Mnih et al. (2013). We show that when the underlying model is not *MDP*, model based *MDP* algorithms learn wrong model representation and model free algorithms, learn Q-function on observation set which does not carry the markovian property anymore. Furthermore, because of non-markovianity on observation set, current observation is not sufficient statistic for policy and sort of memory is required. Assume the game which has one green and one red apples. At the beginning of the game, the emulator reveals a flag which shows which apple has positive reward and which one has negative reward. In this case, the MDP based learner forgets the flag and always suffers regret linear regret.

In this paper, the estimation of the POMDP is carried out via spectral methods which involve decomposition of certain moment tensors computed from data. This learning algorithm is interleaved with the optimization of the planning policy using an exploration–exploitation strategy inspired by the UCRL method for MDPs (Jaksch et al., 2010). The resulting algorithm, called SM-UCRL (*Spectral Method for Upper-Confidence Reinforcement Learning*), runs through epochs of variable length, where the agent follows a fixed policy until enough data are collected and then it updates the current policy according to the estimates of the POMDP parameters and their accuracy.

We derive a regret bound with respect to the best memoryless (stochastic) policy for the given POMDP. Indeed, for a general POMDP, the optimal policy need not be memoryless. However, finding the optimal policy is uncomputable for infinite horizon regret minimization (Madani, 1998). Instead memoryless policies have shown good performance in practice (see the Section on related work). Moreover, for the class of so-called *contextual MDP*, a special class of POMDPs, the optimal policy is also memoryless (Krishnamurthy et al., 2016).

1.2 Related Work

In last few decades, MDP has been widely studied (Kearns and Singh, 2002; Brafman and Tennenholtz, 2003; Jaksch et al., 2010) in different setting. While RL in MDPs has been widely studied, the design of effective exploration–exploration strategies in POMDPs is still relatively unexplored. Ross et al. (2007) and Poupart and Vlassis (2008) propose to integrate the problem of estimating the belief state into a model-based Bayesian RL approach, where a distribution over possible MDPs is updated over time. An alternative to model-based approaches is to adapt model-free algorithms, such as Q-learning, to the case of POMDPs. Perkins (2002) proposes a Monte-Carlo approach to action-value estimation and it shows convergence to locally optimal memoryless policies. An alternative approach to solve POMDPs is to use policy search methods, which avoid estimating value functions and directly optimize the performance by searching in a given policy space, which usually contains memoryless policies (see e.g., (Ng and Jordan, 2000)). Beside its practical success in offline problems, policy search has been successfully integrated with efficient exploration–exploitation techniques and shown to achieve small regret (Gheshlaghi-Azar et al., 2013). Nonetheless, the performance of such methods is severely constrained by the choice of the policy space, which may not contain policies with good performance.

Matrix decomposition methods have been previously used in the more general setting of predictive state representation (PSRs) (Boots et al., 2011) to reconstruct the structure of the dynamical system. Despite the generality of PSRs, the proposed model relies on strong assumptions on the dynamics of the system and it does not have any theoretical guarantee about its performance. Recently, (Hamilton et al., 2014) introduced compressed PSR (CPSR) method to reduce the computation cost in PSR by exploiting the advantages in dimensionality reduction, incremental matrix decomposition, and compressed sensing. In this work, we take these ideas further by considering more powerful tensor decomposition techniques.

In last few decades, latent variable models have become popular model for the problems with

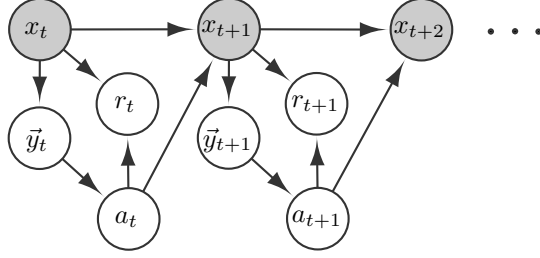


Figure 1: Graphical model of a POMDP under memoryless policies.

partially observable variables. Traditional methods such as Expectation-Maximization (EM) and variational methods have been used to learn the hidden structure of the model but usually they have no consistency guarantees, they are computationally massive, and mostly converge to local optimum which can be arbitrarily bad. To overcome these drawbacks, spectral methods have been used for consistent estimation of a wide class of LVMs Anandkumar et al. (2012), Anandkumar et al. (2014), the theoretical guarantee and computation complexity by using robust tensor power method are well studied in Song et al. (2013) and Wang et al. (2015). Today, spectral methods and tensor decomposition methods are well known as a credible alternative for EM and variational methods for inferring the latent structure of the model. These methods have been shown to be efficient in learning of Gaussian mixture models, topic modeling, Latent Dirichlet Allocation, Hidden Markov model, etc.

2 Preliminaries

A POMDP M is a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{Y}, \mathcal{R}, f_T, f_R, f_O \rangle$, where \mathcal{X} is a finite state space x with cardinality $|\mathcal{X}| = X$, \mathcal{A} is a finite action space a with cardinality $|\mathcal{A}| = A$, \mathcal{Y} is a finite observation space \vec{y} with cardinality $|\mathcal{Y}| = Y$ (the vector representation is w.r.t one-hot encoding), and \mathcal{R} is a finite reward space r with cardinality $|\mathcal{R}| = R$ and largest reward r_{\max} . Finally, f_T denotes the transition density, so that $f_T(x'|x, a)$ is the probability of transition to x' given the state-action pair (x, a) , $\bar{r}(x, a)$ is mean reward at state x and action a . Furthermore, f_O is the observation density, so that $f_O(\vec{y}|x)$ is the probability of receiving the observation in \mathcal{Y} corresponding to the indicator vector \vec{y} given the state x . Whenever convenient, we use tensor forms for the density functions such that

$$T_{i,j,l} = \mathbb{P}[x_{t+1} = j | x_t = i, a_t = l] = f_T(j|i, l), \quad O_{n,i} = \mathbb{P}[\vec{y} = \vec{e}_n | x = i] = f_O(\vec{e}_n|i)$$

Such that $T \in \mathbb{R}^{X \times X \times A}$ and $O \in \mathbb{R}^{Y \times X}$. We also denote by $T_{:,j,l}$ the fiber (vector) in \mathbb{R}^X obtained by fixing the arrival state j and action l and by $T_{:,:,l} \in \mathbb{R}^{X \times X}$ the transition matrix between states when using action l . The graphical model associated to the POMDP is illustrated in Fig. 1.

A policy is stochastic mapping from observations to actions and for any policy π we denote by $f_\pi(a|\vec{y})$ its density function. We denote by \mathcal{P} the set of all stochastic memoryless policies. Acting according to a policy π in a POMDP M defines a Markov chain characterized by a transition density

$$f_{T,\pi}(x'|x) = \sum_a \sum_{\vec{y}} f_\pi(a|\vec{y}) f_O(\vec{y}|x) f_T(x'|x, a),$$

and a stationary distribution ω_π over states such that $\omega_\pi(x) = \sum_{x'} f_{T,\pi}(x'|x) \omega_\pi(x')$. The expected average reward performance of a policy π is

$$\eta(\pi; M) = \sum_x \omega_\pi(x) \bar{r}_\pi(x),$$

where $\bar{r}_\pi(x)$ is the expected reward of executing policy π in state x defined as

$$\bar{r}_\pi(x) = \sum_a \sum_{\vec{y}} f_O(\vec{y}|x) f_\pi(a|\vec{y}) \bar{r}(x, a),$$

The best stochastic memoryless policy in \mathcal{P} is $\pi^+ = \arg \max_{\pi \in \mathcal{P}} \eta(\pi; M)$ and we denote by $\eta^+ = \eta(\pi^+; M)$ its average reward.¹

3 Learning the Parameters of the POMDP

In this section we introduce a novel spectral method to estimate the POMDP parameters f_T , f_O , and f_R . A stochastic policy π is used to generate a trajectory $(\vec{y}_1, a_1, r_1, \dots, \vec{y}_N, a_N, r_N)$ of N steps. Similar to the case of HMMs, the key element to apply the spectral methods is to construct a multi-view model for the hidden states. Despite its similarity, the spectral method developed for HMM by Anandkumar et al. (2014) cannot be directly employed here. In fact, in HMMs the state transition and the observations only depend on the current state. On the other hand, in POMDPs the probability of a transition to state x' not only depends on x , but also on action a . Since the action is chosen according to a memoryless policy π based on the current observation, this creates an indirect dependency of x' on observation \vec{y} , which makes the model more intricate.

3.1 The multi-view model

We estimate POMDP parameters for each action $l \in [A]$ separately. Let $t \in [2, N - 1]$ be a step at which $a_t = l$, we construct three views $(a_{t-1}, \vec{y}_{t-1}, r_{t-1})$, (\vec{y}_t, r_t) , and (\vec{y}_{t+1}) which all contain observable elements. As it can be seen in Fig. 1, all three views provide some information about the hidden state x_t (e.g., the observation \vec{y}_{t-1} triggers the action a_{t-1} , which influence the transition to x_t). A careful analysis of the graph of dependencies shows that conditionally on x_t, a_t all the views are independent. For instance, let us consider \vec{y}_t and \vec{y}_{t+1} . These two random variables are clearly dependent since \vec{y}_t influences action a_t , which triggers a transition to x_{t+1} that emits an observation \vec{y}_{t+1} . Nonetheless, it is sufficient to condition on the action $a_t = l$ to break the dependency and make \vec{y}_t and \vec{y}_{t+1} independent. Similar arguments hold for all the other elements in the views, which can be used to recover the latent variable x_t . More formally, we encode the triple $(a_{t-1}, \vec{y}_{t-1}, r_{t-1})$ into a vector $\vec{v}_{1,t}^{(l)} \in \mathbb{R}^{A \cdot Y}$, so that view $\vec{v}_{1,t}^{(l)} = \vec{e}_s$ whenever $a_{t-1} = k$, $\vec{y}_{t-1} = \vec{e}_n$, and $r_{t-1} = \vec{e}_m$ for a suitable mapping between the index $s \in \{1, \dots, A \cdot Y\}$ and the indices (k, n) of the action, observation, and reward. Similarly, we proceed for $\vec{v}_{2,t}^{(l)} \in \mathbb{R}^Y$ and $\vec{v}_{3,t}^{(l)} \in \mathbb{R}^Y$. We introduce the three view matrices $V_\nu^{(l)}$ with $\nu \in \{1, 2, 3\}$ associated with action l defined as $V_1^{(l)} \in \mathbb{R}^{A \cdot Y \times X}$, $V_2^{(l)} \in \mathbb{R}^{Y \cdot R \times X}$, and $V_3^{(l)} \in \mathbb{R}^{Y \times X}$ such that

$$\begin{aligned} [V_1^{(l)}]_{s,i} &= \mathbb{P}(\vec{v}_{1,t}^{(l)} = \vec{e}_s | x_2 = i) = [V_1^{(l)}]_{(n,m,k),i} = \mathbb{P}(\vec{y}_1 = \vec{e}_n, = \vec{e}_m, a_1 = k | x_2 = i), \\ [V_2^{(l)}]_{s,i} &= \mathbb{P}(\vec{v}_{2,t}^{(l)} = \vec{e}_s | x_2 = i, a_2 = l) = [V_2^{(l)}]_{(n',m'),i} = \mathbb{P}(\vec{y}_2 = \vec{e}_{n'} = \vec{e}_{m'} | x_2 = i, a_2 = l), \\ [V_3^{(l)}]_{s,i} &= \mathbb{P}(\vec{v}_{3,t}^{(l)} = \vec{e}_s | x_2 = i, a_2 = l) = [V_3^{(l)}]_{n'',i} = \mathbb{P}(\vec{y}_3 = \vec{e}_{n''} | x_2 = i, a_2 = l). \end{aligned}$$

At the end, let's concatenat the reward of time $t - 1$ and t to the end of vectors $\vec{v}_{1,t}^{(l)}$ and $\vec{v}_{2,t}^{(l)}$ which ends up to one extra row at the bottom of $V_1^{(l)}$ and $V_2^{(l)}$. By simple manipulation, one can efficiently extract the model parameters out of $V_2^{(l)}$, and $V_3^{(l)}$ for $\forall l \in \mathcal{A}$

Empirical estimates of POMDP parameters. In practice, $M_2^{(l)}$ and $M_3^{(l)}$ are not available and need to be estimated from samples. Given a trajectory of N steps obtained executing policy π , let $\mathcal{T}(l) = \{t \in [2, N - 1] : a_t = l\}$ be the set of steps when action l is played, then we collect all the triples $(a_{t-1}, \vec{y}_{t-1}, r_{t-1})$, (\vec{y}_t, r_t) and (\vec{y}_{t+1}) for any $t \in \mathcal{T}(l)$ and construct the corresponding views $\vec{v}_{1,t}^{(l)}$, $\vec{v}_{2,t}^{(l)}$, $\vec{v}_{3,t}^{(l)}$. Then we symmetrize the views. Given the resulting $\widehat{M}_2^{(l)}$ and $\widehat{M}_3^{(l)}$, we apply the spectral tensor decomposition method to recover empirical estimates of second and third views $\widehat{V}_2^{(l)}$, $\widehat{V}_3^{(l)}$. Thereafter, a simple manipulation results in the model parameters. The overall method is summarized in Alg. 1. The empirical estimates of the POMDP parameters enjoy the following guarantee.

¹We use π^+ rather than π^* to recall the fact that we restrict the attention to \mathcal{P} and the actual optimal policy for a POMDP in general should be constructed on the belief-MDP.

Algorithm 1 Estimation of the POMDP parameters. The routine TENSORDECOMPOSITION refers to the spectral tensor decomposition method of Azzadene et al. (2016b).

Input:

Policy density f_π , number of states X
Trajectory $\langle (\vec{y}_1, a_1, r_1), (\vec{y}_2, a_2, r_2), \dots, (\vec{y}_N, a_N, r_N) \rangle$

Variables:

Estimated second and third views $\widehat{V}_2^{(l)}$, and $\widehat{V}_3^{(l)}$ for any action $l \in [A]$
Estimated observation, reward, and transition models $\widehat{f}_O, \widehat{r}, \widehat{f}_T$

for $l = 1, \dots, A$ **do**

Set $\mathcal{T}(l) = \{t \in [N-1] : a_t = l\}$ and $N(l) = |\mathcal{T}(l)|$

Construct views $\vec{v}_{1,t}^{(l)} = (a_{t-1}, \vec{y}_{t-1}, r_{t-1})$, $\vec{v}_{2,t}^{(l)} = (\vec{y}_t, r_t)$, $\vec{v}_{3,t}^{(l)} = \vec{y}_{t+1}$ for any $t \in \mathcal{T}(l)$

Symmetrize the views

Compute second and third moments

$$\widehat{M}_2^{(l)} = \frac{1}{N(l)} \sum_{t \in \mathcal{T}_l} \widetilde{v}_{1,t}^{(l)} \otimes \widetilde{v}_{2,t}^{(l)}, \quad \widehat{M}_3^{(l)} = \frac{1}{N(l)} \sum_{t \in \mathcal{T}_l} \widetilde{v}_{1,t}^{(l)} \otimes \widetilde{v}_{2,t}^{(l)} \otimes \widetilde{v}_{3,t}^{(l)}$$

Compute $\widehat{V}_3^{(l)} = \text{TENSORDECOMPOSITION}(\widehat{M}_2^{(l)}, \widehat{M}_3^{(l)})$

end for

Estimate $\widehat{f}_O, \widehat{r}, \widehat{f}_T$

Return: $\widehat{r}, \widehat{f}_T, \widehat{f}_O, \mathcal{B}_R, \mathcal{B}_T, \mathcal{B}_O$

Theorem 1 (Learning Parameters) Let $\widehat{f}_O, \widehat{f}_T$, and \widehat{f}_R be the estimated POMDP models using a trajectory of N steps. we have

$$\|\widehat{f}_O(\cdot|i) - f_O(\cdot|i)\|_1 \leq \mathcal{B}_O^{(l)} := C_O \sqrt{\frac{Y \log(1/\delta)}{N(l)}}, \quad \|\widehat{f}_R(\cdot|i, l) - f_R(\cdot|i, l)\|_1 \leq \mathcal{B}_R^{(l)} := C_R \sqrt{\frac{\log(1/\delta)}{N(l)}}, \quad (1)$$

$$\|\widehat{f}_T(\cdot|i, l) - f_T(\cdot|i, l)\|_2 \leq \mathcal{B}_T^{(l)} := C_T \sqrt{\frac{Y X^2 \log(1/\delta)}{N(l)}}, \quad (2)$$

with probability $1 - 6(Y^2 + AY)A\delta$ (w.r.t. the randomness in the transitions, observations, and policy), where C_O, C_R , and C_T are numerical constants.

Remark 1 (consistency and dimensionality). All previous errors decrease with a rate $\widetilde{O}(1/\sqrt{N(l)})$, showing the consistency of the spectral method, so that if all the actions are repeatedly tried over time, the estimates converge to the true parameters of the POMDP. This is in contrast with EM-based methods which typically get stuck in local maxima and return biased estimators, thus preventing from deriving confidence intervals.

4 Spectral UCRL

The most interesting aspect of the estimation process illustrated in the previous section is that it can be applied when samples are collected using any policy π in the set \mathcal{P} . As a result, it can be integrated into any exploration-exploitation strategy where the policy changes over time in the attempt of minimizing the regret.

The algorithm. The SM-UCRL algorithm illustrated in Alg. 2 is the result of the integration of the spectral method into a structure similar to UCRL (Jaksch et al., 2010) designed to optimize the exploration-exploitation trade-off. The learning process is split into episodes of increasing length. At the beginning of each episode $k > 1$ (the first episode is used to initialize the variables), an estimated POMDP $\widehat{M}^{(k)} = (X, A, Y, R, \widehat{f}_T^{(k)}, \widehat{f}_R^{(k)}, \widehat{f}_O^{(k)})$ is computed using the spectral method of Alg. 1.

Algorithm 2 The SM-UCRL algorithm.

Input: Confidence δ'

Variables:

Number of samples $N^{(k)}(l)$

Estimated observation, reward, and transition models $\widehat{f}_O^{(k)}, \widehat{f}_R^{(k)}, \widehat{f}_T^{(k)}$

Initialize: $t = 1$, initial state x_1 , $\delta = \delta'/N^6$, $k = 1$

while $t < N$ **do**

 Compute the estimated POMDP $\widehat{M}^{(k)}$ with the Alg. 1 using $N^{(k)}(l)$ samples per action

 Compute the set of admissible POMDPs $\mathcal{M}^{(k)}$ using bounds in Thm. 1

 Compute the optimistic policy $\widetilde{\pi}^{(k)} = \arg \max_{\pi \in \mathcal{P}} \max_{M \in \mathcal{M}^{(k)}} \eta(\pi; M)$

 Set $v^{(k)}(l) = 0$ for all actions $l \in [A]$

while $\forall l \in [A], v^{(k)}(l) < 2N^{(k)}(l)$ **do**

 Execute $a_t \sim f_{\widetilde{\pi}^{(k)}}(\cdot | \vec{y}_t)$

 Obtain reward \vec{r}_t , observe next observation \vec{y}_{t+1} , and set $t = t + 1$

end while

 Store $N^{(k+1)}(l) = \max_{k' \leq k} v^{(k')}(l)$ samples for each action $l \in [A]$

 Set $k = k + 1$

end while

Given the estimated POMDP $\widehat{M}^{(k)}$ and the result of Thm. 1, we construct the set $\mathcal{M}^{(k)}$ of *admissible* POMDPs $\widetilde{M} = \langle \mathcal{X}, \mathcal{A}, \mathcal{Y}, \mathcal{R}, \widetilde{f}_T, \widetilde{f}_R, \widetilde{f}_O \rangle$ whose transition, reward, and observation models belong to the confidence intervals and compute the optimal policy with respect to optimistic model as follows

$$\widetilde{\pi}^{(k)} = \arg \max_{\pi \in \mathcal{P}} \max_{M \in \mathcal{M}^{(k)}} \eta(\pi; M). \quad (3)$$

The choice of using the optimistic POMDP guarantees the $\widetilde{\pi}^{(k)}$ explores more often actions corresponding to large confidence intervals, thus contributing to improve the estimates over time. After computing the optimistic policy, $\widetilde{\pi}^{(k)}$ is executed until the number of samples for one action is doubled, i.e., $v^{(k)}(l) \geq 2N^{(k)}(l)$. This stopping criterion avoids switching policies too often and it guarantees that when an epoch is terminated, enough samples are collected to compute a new (better) policy. This process is then repeated over epochs and we expect the optimistic policy to get progressively closer to the best policy $\pi^+ \in \mathcal{P}$ as the estimates of the POMDP get more and more accurate.

Regret analysis. We now study the regret SM-UCRL w.r.t. the best policy in \mathcal{P} . Given an horizon of N steps, the regret is defined as

$$\text{Reg}_N = N\eta^+ - \sum_{t=1}^N r_t, \quad (4)$$

where r_t is the random reward obtained at time t over the states traversed by the policies performed over epochs on the actual POMDP. To restate, the complexity of learning in a POMDP M is partially determined by its diameter, defined as

$$D := \max_{x, x' \in \mathcal{X}, a, a' \in \mathcal{A}} \min_{\pi \in \mathcal{P}} \mathbb{E}[\tau(x', a' | x, a; \pi)], \quad (5)$$

which corresponds to the expected passing time from a state x to a state x' starting with action a and terminating with action a' and following the most effective memoryless policy $\pi \in \mathcal{P}$. Now for result we have the following theorem.

Theorem 2 (Regret Bound) *Consider a POMDP M with X states, A actions, Y observations, R rewards, characterized by a diameter D . If SM-UCRL is run over N steps and the confidence intervals of Thm. 1 are used with $\delta = \delta'/N^6$ in constructing the plausible POMDPs \widehat{M} , then the total regret*

$$\text{Reg}_N \leq C_1 r_{\max} D X^{3/2} \sqrt{AYN \log(N/\delta')} \quad (6)$$

with probability $1 - \delta'$, where C_1 is numerical constants

5 Experiments

In the following section, we show how SM-UCRL algorithm outperforms other well-known methods in both synthetic environment and simple computer game.

5.1 Synthetic Environment

In this subsection, we illustrate the performance of our method on a simple synthetic environment which follows a POMDP structure with $X = 2, Y = 4, A = 2, R = 4$, and $r_{max} = 4$. We find that spectral learning method quickly learn model parameters Fig. [2]. Estimation of the transition tensor T takes more effort compared to estimation of observation matrix O and reward matrix R due to the fact that the transition tensor is estimated given estimated O matrix which adds up more error. For planning, given POMDP model parameters, we use alternating maximization method to find the memoryless policy. This method, iteratively, alternates between updates of the policy and the stationary distribution which ends up to stationary point of the optimization problem. We find that, in practice, this method converges to a reasonably good solution (Azizzadenesheli et al. (2016a) shows the planing is NP-hard in general). The resulting regret bounds are shown in Fig. [5.1]. We compare against the following algorithm: (1) baseline random policies which simply selects random actions without looking at the observed data, (2) UCRL-MDP Auer et al. (2009) which attempts to fit a MDP model to the observed data and runs the UCRL policy, and (3) Q-Learning Watkins and Dayan (1992) which is a model-free method that updates policy based on the Q-function. We find that our method converges much faster. In addition, we show that it converges to a much better policy (stochastic). Note that the MDP-based policies UCRL-MDP and Q-Learning perform very poorly, and are even worse than a random policy and are too far from SM-UCRL policy. This is due to model misspecification and dealing with larger state space.

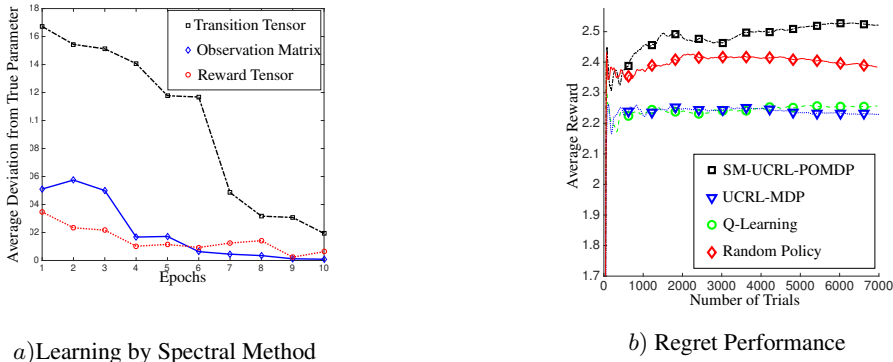


Figure 2: (a)Spectral estimation error Eqs. [1,2].(b) Average reward Comparison.

5.2 Simple Atari Game

In the following, we provide empirical guarantees for SM-UCRL on environment of simple computer Game. In this game, Figs. [3], the environment is 10×10 grid world and has five sweet (*green*) apple and five poisonous apples (*red*). The environment uniformly spread out the apples. In addition, each of these apples lasts uniformly for (1, 2) time steps and disappear or they get eaten by the agent. In this game, the agent interacts with this environment. We study two settings, (*i*) the has set of possible actions ((N, W, S, E)), (*ii*) set of actions ($(N, NW, W, SW, S, SE, E, NE)$). At each time step the agent chooses an action and deterministically moves one step in that direction. If there is a sweet apple at the new location, the agent will score up by one, and score down by one if it is poisonous one. In this game, at each time step, the agent just partially observes the environment, Fig. [3,a] one single box above of the agent is visible to her, and Fig. [3,b] three boxes above of her are observable. The randomness on rewarding process and partial observability bring the notion of hidden structure and pushes the environment more to be POMDPs models rather than MDPs.

For single box observable setting, the observation set has cardinality of 4, (*wall, sweet apple, poisonous apple, nothing*). We tune SM-UCRL with ($X=3$), since it is not known (we add minimum level of stochasticity to the policy when the policy suggests deterministic mapping to an action).

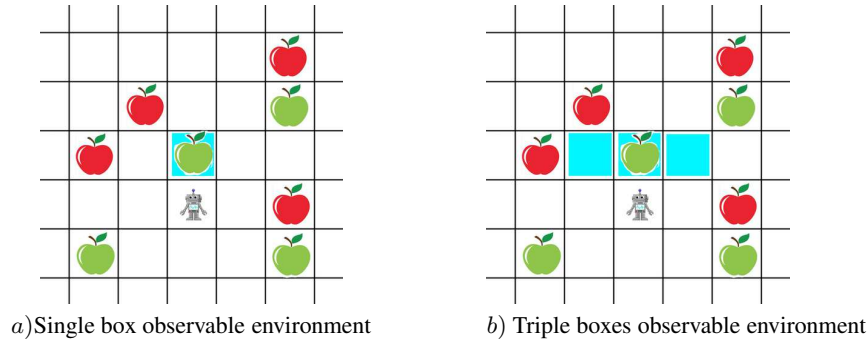


Figure 3: Environment configuration

In addition, we apply DQN (Deep Q-Network) Mnih et al. (2013) with three hidden layers and 10 hidden units of *hyperbolic tangent* activation functions in each hidden layer. For back propagation, we use *RMSProp* method which has been shown to be robust and stable. Figs. [4] shows the performance of both SM-UCRL and DQN (DNN) for both case when action set is (N, W, S, E) and $(N, NW, W, SW, S, SE, E, NE)$. We show that not only SM-UCRL captures the environment behavior faster but also reaches to the better long term average reward. We run DQN couple of times and represent the average performance as it is shown in Figs. [5.2,5.2]. DQN some times traps in some local minima and it results in bad performance which degrades its average performance.

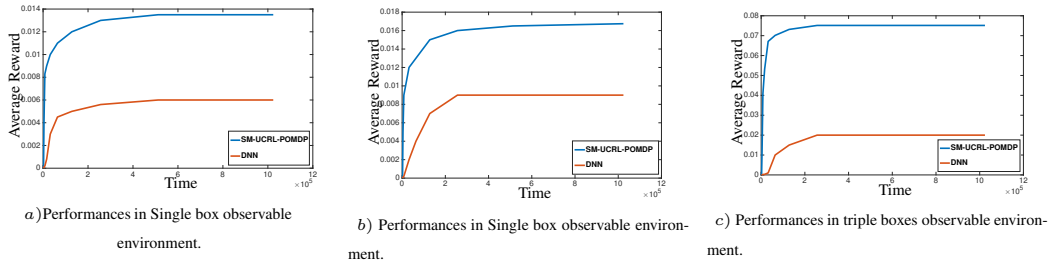


Figure 4: (a) Action set (N, W, S, E) , (b,c) Action set $(N, NW, W, SW, S, SE, E, NE)$.

In other setting, when three boxes are observable Fig. [3,b], the observation set has cardinality of 64 (four possible observation for each of these three boxes). We tune the SM-UCRL with $(X=8)$ and apply it on this environment. Again, we show that SM-UCRL outperforms DQN with same structure except 30 hidden units in each hidden layer. During the implementation, we observed that SM-UCRL does not need to estimate the model parameter very well to get to reasonable policy. It comes up with the stochastic and reasonably good policy even from the beginning. On the other hand, we observed that the policy makes balance between moving upward, downward and makes a good balance between moving rightward and leftward in order to keep the agent away from the walls. It helps the agent to collect more reward and move around the area far from the walls. (link:<https://newport.eecs.uci.edu/anandkumar/pubs/SMvsDQN.flv>)

6 Conclusion

We introduced a novel RL algorithm for POMDPs which relies on a spectral method to consistently identify the parameters of the POMDP and an optimistic approach for the solution of the exploration-exploitation problem. For the resulting algorithm we derive confidence intervals on the parameters and a minimax optimal bound for the regret.

This work opens several interesting directions for future development. POMDP is a special case of the predictive state representation (PSR) model Littman et al. (2001), which allows representing more sophisticated dynamical systems. Given the spectral method developed in this paper, a natural extension is to apply it to the more general PSR model and integrate it with an exploration-exploitation algorithm to achieve bounded regret. As long as POMDPs are more suitable models for most of real world applications compared to MDP, the further experimental analyses are interesting.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832.

- Anandkumar, A., Hsu, D., and Kakade, S. M. (2012). A method of moments for mixture models and hidden markov models. *arXiv preprint arXiv:1203.0683*.
- Auer, P., Jaksch, T., and Ortner, R. (2009). Near-optimal regret bounds for reinforcement learning. In *Advances in neural information processing systems*, pages 89–96.
- Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. (2016a). Open problem: Approximate planning of pomdps in the class of memoryless policies. *arXiv preprint arXiv:1608.04996*.
- Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. (2016b). Reinforcement learning of pomdps using spectral methods. *arXiv preprint arXiv:1602.07764*.
- Boots, B., Siddiqi, S. M., and Gordon, G. J. (2011). Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966.
- Brafman, R. I. and Tenenbaum, M. (2003). R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231.
- Gheshlaghi-Azar, M., Lazaric, A., and Brunskill, E. (2013). Regret bounds for reinforcement learning with policy advice. In *Proceedings of the European Conference on Machine Learning (ECML’13)*.
- Hamilton, W., Fard, M. M., and Pineau, J. (2014). Efficient learning and planning with compressed predictive states. *The Journal of Machine Learning Research*, 15(1):3395–3439.
- Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, 11:1563–1600.
- Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232.
- Krishnamurthy, A., Agarwal, A., and Langford, J. (2016). Contextual-mdps for pac-reinforcement learning with rich observations. *arXiv preprint arXiv:1602.02722v1*.
- LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.
- Littman, M. L., Sutton, R. S., and Singh, S. (2001). Predictive representations of state. In *In Advances In Neural Information Processing Systems 14*, pages 1555–1561. MIT Press.
- Madani, O. (1998). On the computability of infinite-horizon partially observable markov decision processes. In *AAAI98 Fall Symposium on Planning with POMDPs, Orlando, FL*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Ng, A. Y. and Jordan, M. (2000). Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, UAI’00*, pages 406–415, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Papadimitriou, C. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450.
- Perkins, T. J. (2002). Reinforcement learning for POMDPs based on action values and stochastic optimization. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2002)*, pages 199–204. AAAI Press.
- Poupart, P. and Vlassis, N. (2008). Model-based bayesian reinforcement learning in partially observable domains. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*.
- Ross, S., Chaib-draa, B., and Pineau, J. (2007). Bayes-adaptive pomdps. In *Advances in neural information processing systems*, pages 1225–1232.
- Sondik, E. J. (1971). *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University.
- Song, L., Anandkumar, A., Dai, B., and Xie, B. (2013). Nonparametric estimation of multi-view latent variable models. *arXiv preprint arXiv:1311.3287*.
- Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*. MIT Press.
- Wang, Y., Tung, H.-Y., Smola, A. J., and Anandkumar, A. (2015). Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems*, pages 991–999.
- Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.