2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Hidden Markov Model Estimation-Based Q-learning for Partially Observable Markov Decision Process

Hyung-Jin Yoon, Donghwan Lee, and Naira Hovakimyan

Abstract— **The objective is to study an on-line Hidden Markov model (HMM) estimation-based Q-learning algorithm for partially observable Markov decision process (POMDP) on finite state and action sets. When the full state observation is available, Q-learning finds the optimal action-value function given the current action (Q-function). However, Q-learning can perform poorly when the full state observation is not available. In this paper, we formulate the POMDP estimation into a HMM estimation problem and propose a recursive algorithm to estimate both the POMDP parameter and Q-function concurrently. Also, we show that the POMDP estimation converges to a set of stationary points for the maximum likelihood estimate, and the Q-function estimation converges to a fixed point that satisfies the Bellman optimality equation weighted on the invariant distribution of the state belief determined by the HMM estimation process.**

I. Introduction

Reinforcement learning (RL) is getting significant attention due to the recent successful demonstration of the 'Go game', where the RL agents outperform humans in certain tasks (video game [1], playing Go [2]). Although the demonstration shows the great potential of the RL, those game environments are confined and restrictive compared to what ordinary humans go through in their everyday life. One of the major differences between the game environment and the real-life is the presence of unknown factors, i.e. the observation of the state of the environment is incomplete. Most RL algorithms are based on the assumption that complete state observation is available, and the state transition depends on the current state and the action (Markovian assumption). Markov decision process (MDP) is a modeling framework with the Markovian assumption. Development and analysis of the standard RL algorithm are based on MDP. Applying those RL algorithms with incomplete observation may lead to poor performance. In [3], the authors showed that a standard policy evaluation algorithm can result in an arbitrary error due to the incomplete state observation.

Partially observable Markov decision process (POMDP) is a generalization of MDP that incorporates the incomplete state observation model. When the model parameter of a POMDP is given, the optimal policy is determined by using dynamic programming on the belief state of MDP, which is transformed from the POMDP [4]. The belief state of MDP has continuous state space, even though the corresponding POMDP has finite state space. Hence, solving a dynamic programming problem on the belief state of MDP is computationally challenging. There exists a number of results to obtain approximate solutions to the optimal policy, when the model is given, [4]–[6]. When the model of POMDP is not given (model-free), a choice is in the policy gradient approach without relying on Bellman's optimality. However, the policy gradient estimate has high variance so that convergence to the optimal policy typically takes longer as compared to other RL algorithms, which use Bellman's optimality principle.

In this paper, we aim to develop a recursive estimation algorithm for a POMDP to estimate the parameters of the model, predict the hidden state, and also determine the optimal value state function concurrently. The idea of using a recursive state predictor (Bayesian state belief filter) in RL was investigated in [7]–[10]. However, the algorithms in [7]–[9] require the POMDP model parameter knowledge[1]. A parameter-free reinforcement learning that uses HMM formulation is presented in [10]. The result in [10] shares the same idea as ours, where we use HMM estimator with a fixed behavior policy, in order to disambiguate the hidden state, learn the POMDP parameters, and find optimal policy. However, the algorithm in [10] involves multiple phases, including identification and design, which are hard to apply online to real-time learning tasks, whereas recursive estimation is more suitable (e.g., DQN, DDPG, or Q-learning are online algorithms). The main contribution of this paper is to present and analyze a new on-line estimation algorithm to simultaneously estimate the POMDP model parameters and corresponding optimal action-value function (Q-function), where we employ online HMM estimation techniques [11], [12].

The remainder of the paper is organized as follows. In Section II, HMM interpretation of the POMDP with a behavior policy is presented. In Section III, the proposed recursive estimation of the HMM, POMDP, and Q-function is presented, and the convergence of the estimator is analyzed. In Section IV, a numerical example is presented. Section V summarizes the paper.

II. A HMM: POMDP excited by Behavior policy

We consider a partially observable Markov decision process (POMDP) on finite state and action sets. A fixed behavior policy excites the POMDP so that all pairs of state-action are realized infinitely often.

Research supported by NSF NRI initiative #1528036 and #1830639.

Hyung-Jin Yoon and Naira Hovakimyan are with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801, USA. Donghwan Lee is with the Department of Industrial and Enterprise Systems Engineering in UIUC. {hyoon33, nhovakim, donghwan}@illinois.edu

[1] In [8], the algorithm needs full state observation for the system identification of POMDP.

A. POMDP on finite state-action sets

The POMDP $(\mathcal{S}, \mathcal{A}, T_a(s,s'), R(s,a), \mathcal{O}, O(o,s), \gamma)$ comprises: a finite state space $\mathcal{S} := \{1,\dots,I\}$, a finite action space $\mathcal{A} := \{1,\dots,K\}$, a state transition probability $T_a(s,s') = P(s_{n+1}=s'|s_n=s, a_n=a)$, for $s,s' \in \mathcal{S}$ and $a \in \mathcal{A}$, a reward model $R \in \mathbb{R}$ such that $R(s,a) = r(s,a) + \delta$, where δ denotes independent identically distributed (i.i.d.) Gaussian noise $\delta \sim \mathcal{N}(0,\sigma^2)$, a finite observation space $\mathcal{O} := \{1,\dots,J\}$, an observation probability $O(o,s) = P(o_n = o|s_n = s)$, and the discount factor $\gamma \in [0,1)$. At each time step n, the agent first observes $o_n \in \mathcal{O}$ from the environment at the state $s_n \in \mathcal{S}$, does action $a_n \in \mathcal{A}$ on the environment and gets the reward $r_n \in \mathbb{R}$ in accordance to $R(s,a)$.

B. Behavior policy and HMM

The behavior policy's purpose is system identification (in other words, estimation of the POMDP parameter). We denote the behavior policy by μ, which is a conditional probability, i.e. $\mu(o) = P(a|o)$. The POMDP with $\mu(o)$ becomes a hidden Markov model (HMM), as illustrated in Fig. 1.

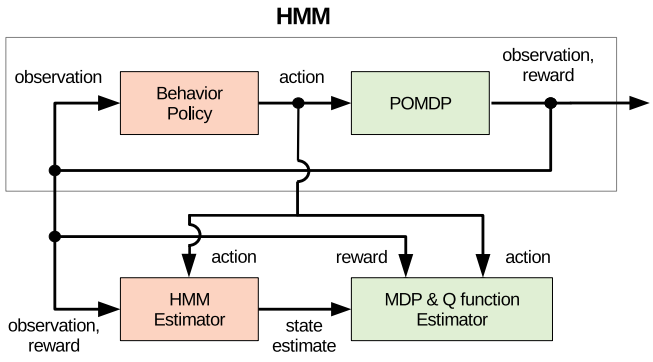

Fig. 1: A POMDP Estimation Framework.

The HMM comprises: state transition probability $P(s_{n+1}=s'|s_n=s) = P(s_{n+1}=s'|s_n=s, a_n=a; \mu, O)$ for all pairs of (s,s') and the extended observation probability, i.e. $P(o,a,r|s)$ that is determined by the POMDP model parameters: $O(o,s)$, $R(s,a)$ and the behavior policy $\mu(o)$.

For the ease of notation, we define the following tensor and matrices: $\mathbf{T} \in \mathbb{R}^{K\times I\times I}$ such that $\mathbf{T}_{ijk} = P(s_{n+1} = k|s_n = j, a_n = i)$, $\mathbf{R} \in \mathbb{R}^{K\times I}$ such that $\mathbf{R}_{ij} = r(s=j, a=i)$, $\mathbf{O} \in \mathbb{R}^{I\times J}$ such that $\mathbf{O}_{ij} = P(o_n = j|s_n = i)$, and $\mathbf{P} \in \mathbb{R}^{I\times I}$ such that $\mathbf{P}_{ij} = P(s_{n+1}=j|s_n=j;\mu)$.

The HMM estimator in Fig. 1 learns the model parameters $\mathbf{P}, \mathbf{O}, \mathbf{R}, \sigma$, and also provides the state estimate (or belief state) to the MDP and Q-function estimator. Given the transition of the state estimates and the action, the MDP estimator learns the transition model parameter $\mathbf{T}$. Also, the optimal action-value function $Q^*(s,a)$ is recursively estimated based on the transition of the state estimates, reward sample and the action taken.

Algorithm 1 HMM Q-Learning

1: Set $n = 0$.
2: Observe o_0 from the environment.
3: Initialize: the parameter (θ_0, Q_0, T_0), the states $(\mathbf{u}_0, \omega_0)$, $\hat{p}_n^{(\text{prev})} \in \mathcal{P}(\mathcal{S})$ as uniform distribution, randomly choose $a_n^{(\text{prev})} \in \mathcal{A}$, and set $r_n^{(\text{prev})} = 0$.
4: **repeat**
5: Act a with $\mu(o_n) = P(a|o_n)$, get reward r and the next observation o' from the environment.
6: Use $y_n = (o_n, a, r)$ and $(\theta_n, \mathbf{u}_n, \omega_n)$ to update the estimator as follows:

$$\theta_{n+1} = \Pi_H\left[\theta_n + \epsilon_n \mathbf{S}\left(y_n, \mathbf{u}_n, \omega_n; \theta_n\right)\right],$$
$$\mathbf{u}_{n+1} = f(y_n, \mathbf{u}_n; \theta_n),$$
$$\omega_{n+1}^{(l)} = \Phi(y_n, \mathbf{u}_n; \theta_n)\omega_n^{(l)} + \frac{\partial f(y_n, \mathbf{u}_n; \theta_n)}{\partial \theta^{(l)}},$$

where

$$f(y_n, \mathbf{u}_n; \theta_n) \triangleq \frac{\mathbf{P}_{\theta_n}^\top \mathbf{B}(y_n; \theta_n)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n},$$
$$\mathbf{S}\left(y_n, \mathbf{u}_n, \omega_n; \theta_n\right) = \frac{\partial \log\left(\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n\right)}{\partial \theta},$$

Π_H denotes the projection on the convex constraint set $H \subseteq \Theta$, $\epsilon_n \geq 0$ denotes the step size, $\omega_n \in \mathbb{R}^{I\times L}$ denotes the Jacobian of the state prediction vector $\mathbf{u}_n$ with respect to the parameter vector θ_n.
7: Calculate $\hat{p}_n := [P(s=i|y_n, \mathbf{u}_n; \theta_n)]_{i\in\mathcal{I}}$ as in (15).
8: Calculate $\hat{p}(s_{n-1}, s_n)$ with $\hat{p}_n^{(\text{prev})}$ and $\hat{p}_n$ as in (14).
9: Use r_n^{prev}, a_n^{prev} and $\hat{p}(s_{n-1}, s_n)$ to update Q_n according to (16).
10: Use $\hat{p}(s_{n-1}, s_n)$ to update T_n according to (18).
11: $(\hat{p}_n^{(\text{prev})}, r_n^{\text{prev}}, a_n^{\text{prev}}) \leftarrow (\hat{p}_n, \text{r}, \text{a})$.
12: $o_n \leftarrow o'$.
13: $n \leftarrow n+1$.
14: **until** a certain stopping criterion is satisfied.

III. HMM Q-Learning Algorithm For POMDPs

We present a new HMM model estimation-based Q-learning algorithm, called HMM Q-learning, for POMDPs. The pseudo code of the recursive algorithm is in Algorithm 1.

The recursive algorithm integrates (a) the HMM estimation, (b) MDP transition model estimation, and (c) the Q-function estimation steps. Through the remaining subsections, we prove the convergence of Algorithm 1. To this end, we first make the following assumptions.

***Assumption** 1:* The transition probability matrix $\mathbf{P}$ determined by the transition $\mathbf{T}$, the observation $\mathbf{O}$, and the behavior policy $\mu(o)$ are aperiodic and irreducible [13]. Furthermore, we assume that the state-action pair visit probability is strictly positive under the behavior policy.

We additionally assume the following.

***Assumption** 2:* All elements in the observation probability matrix $\mathbf{O}$ are strictly positive, i.e. $\mathbf{O}_{i,j} > 0$ for all $i \in \mathcal{S}$

and $j \in \mathcal{O}$.

Under these assumptions, we will prove the following convergence result.

***Proposition** 1 (Main convergence result):* Suppose that Assumption 1 and Assumption 2 hold. Then the following statements are true:

(i) The iterate θ_n in Algorithm 1 converges almost surely to the stationary point θ^* of the conditional log-likelihood density function based on the sequence of the extended observations $\{y_i = (o_i, r_i, a_i)\}_{i=0}^n$, $l_n(\theta) = \frac{1}{n+1} \log p_n(y_0, y_1, \ldots, y_n | s_0, s_1, \ldots, s_n; \theta)$, i.e., the point θ is satisfying

$$E\left[\frac{\partial \log\left(\mathbf{b}^\top(y_n;\theta)\mathbf{u}_n\right)}{\partial\theta}\right] \in N_H(\theta),$$

where $N_H(\theta)$ is the normal cone [14, pp. 343] of the convex set H at $\theta \in H$, and the expectation E is taken with respect to the invariant distribution of y_n and $\mathbf{u}_n$.

(ii) Define $\bar{p}(s, s') := \lim_{n\to\infty} \hat{p}(s_{n-1}, s_n)$ in the almost sure convergence sense. Then the iterate $\{Q_n\}$ in Algorithm 1 converges in distribution to the optimal Q-function $\hat{Q}^*$, satisfying

$$\hat{Q}^*(s,a) = \sum_{s'} \bar{p}(s,s')\left(r(s,a) + \gamma \max_{a'} \hat{Q}^*(s',a')\right).$$

A. HMM Estimation

We employ the recursive estimators of HMM from [11], [12] for our estimation problem, where we estimate the true parameter θ^* with the model parameters $(\mathbf{P}, \mathbf{R}, \mathbf{O}, \sigma)$ being parametrized as continuously differentiable functions of the vector of real numbers $\theta \in \Theta \subset \mathbb{R}^L$, such that $\theta^* \in \Theta$ and $(\mathbf{P}_{\theta^*}, \mathbf{R}_{\theta^*}, \mathbf{O}_{\theta^*}, \sigma_{\theta^*}) = (\mathbf{P}, \mathbf{R}, \mathbf{O}, \sigma)$. We denote the functions of the parameter as $(\mathbf{P}_\theta, \mathbf{R}_\theta, \mathbf{O}_\theta, \sigma_\theta)$ respectively. In this paper, we consider the normalized exponential function (or softmax function)[2] to parametrize the probability matrices $\mathbf{P}_\theta$, $\mathbf{O}_\theta$. The reward matrix $\mathbf{R}_\theta$ is a matrix in $\mathbb{R}^{I\times K}$, and σ_θ is a scalar.

The iterate θ_n of the recursive estimator converges to the set of the stationary points, where the gradient of the log-likelihood density function is zero [11], [12]. The conditional log-likelihood density function based on the sequence of the extended observations $\{y_i = (o_i, r_i, a_i)\}_{i=0}^n$ is

$$l_n(\theta) = \frac{1}{n+1} \log p_n(y_0, y_1, \ldots, y_n | s_0, s_1, \ldots, s_n; \theta). \quad (1)$$

When the state transition and observation model parameters are available, the state estimate

$$\mathbf{u}_n = [u_{n,1}, u_{n,2}, \ldots, u_{n,I}]^\top, \quad (2)$$

where $u_{n,i} = P(s_n = i | y_0, y_1, \ldots, y_n; \theta)$ is calculated from the recursive state predictor (Bayesian state belief filter) [15]. The state predictor is given as follows:

$$\mathbf{u}_{n+1} = \frac{\mathbf{P}_\theta^\top \mathbf{B}(y_n;\theta)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta)\mathbf{u}_n}, \quad (3)$$

[2]Let $\{\alpha_{1,1}, \ldots, \alpha_{I,I}\}$ denote the parameters for the probability matrix $\mathbf{P}_\theta$. Then the $(i,j)^{\text{th}}$ element of $\mathbf{P}_\theta$ is $\frac{\exp(\alpha_{i,j})}{\sum_{j'=1}^I \exp(\alpha_{i,j'})}$.

where

$$\mathbf{b}(y_n;\theta) = [b_1(y_n;\theta), b_2(y_n;\theta), \ldots, b_I(y_n;\theta)]^\top, \quad (4)$$

$$\begin{aligned} b_i(y_n;\theta) &= p(y_n | s_n = i;\theta) \\ &= P(o_n | s_n = i;\theta) P(a_n | o_n) p(r_n | s_n = i, a_n;\theta), \end{aligned}$$

and $\mathbf{B}(y_n;\theta)$ is the diagonal matrix with $\mathbf{b}(y_n;\theta)$. Using Markov property of the state transitions and the conditional independence of the observations given the states, the conditional likelihood density (1) is written as follows:

$$l_n(\theta) = \frac{1}{n+1} \sum_{k=0}^{n} \log\left(\mathbf{b}^\top(y_n;\theta)\mathbf{u}_n\right). \quad (5)$$

We first introduce the HMM estimator [11], [12] and then apply the convergence result [11] to our estimation task. The recursive HMM estimation in Algorithm 1 is given by:

$$\theta_{n+1} = \Pi_H\left[\theta_n + \epsilon_n \mathbf{S}\left(y_n, \mathbf{u}_n, \omega_n; \theta_n\right)\right], \quad (6)$$

$$\mathbf{S}\left(y_n, \mathbf{u}_n, \omega_n; \theta_n\right) = \frac{\partial \log\left(\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n\right)}{\partial\theta}, \quad (7)$$

where Π_H denotes the projection onto the convex constraint set $H \subseteq \Theta$, $\epsilon_n \geq 0$ denotes the diminishing step-size such that $\epsilon_n \to 0$, $\sum_n \epsilon_n = \infty$, $\omega_n \in \mathbb{R}^{I\times L}$ denotes the Jacobian of the state prediction vector $\mathbf{u}_n$ with respect to the parameter vector θ_n.

Equation (7) can be written in terms of $\mathbf{u}_n$, ω_n, $\mathbf{b}(y_n;\theta_n)$, and its partial derivatives as follows:

$$\begin{aligned} &S^{(l)}\left(y_n, \mathbf{u}_n, \omega_n; \theta_n\right) \\ &= \frac{\mathbf{b}^\top(y_n;\theta_n)\omega_n^{(l)}}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n} + \frac{\left(\left(\partial/\partial\theta^{(l)}\right)\mathbf{b}^\top(y_n;\theta_n)\right)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n}, \end{aligned} \quad (8)$$

where the superscript l of $S^{(l)}(\cdot)$ denotes the l^{th} element of $\mathbf{S}(\cdot)$, and $\omega_n^{(l)}$ is the l^{th} column of the $\omega_n \in \mathbb{R}^{I\times L}$, $\mathbf{u}_n(\theta_n)$ is recursively updated using the state predictor in (3) as

$$\mathbf{u}_{n+1} = \frac{\mathbf{P}_{\theta_n}^\top \mathbf{B}(y_n;\theta_n)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n} \triangleq f(y_n, \mathbf{u}_n;\theta_n), \quad (9)$$

with $\mathbf{u}_0$ being initialized as an arbitrary distribution on the finite state set, $\mathbf{P}_{\theta_n}$ being the state transition probability matrix for the current iterate θ_n. The predicted state estimate is used recursively to calculate the state prediction in the next step. Taking derivative on the update law (9), the update law for $\omega_n^{(l)}$ is

$$\omega_{n+1}^{(l)} = \Phi(y_n, \mathbf{u}_n;\theta_n)\omega_n^{(l)} + \frac{\partial f(y_n, \mathbf{u}_n;\theta_n)}{\partial\theta^{(l)}}, \quad (10)$$

where

$$\Phi(y_n, \mathbf{u}_n;\theta_n) = \frac{\mathbf{P}_{\theta_n}^\top \mathbf{B}(y_n;\theta_n)}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n}\left(\mathbf{I} - \frac{\mathbf{u}_n\mathbf{b}^\top(y_n;\theta_n)}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n}\right),$$

$$\begin{aligned} &\frac{\partial f(y_n, \mathbf{u}_n;\theta_n)}{\partial\theta^{(l)}} \\ &= \mathbf{P}_{\theta_n}^\top\left(\mathbf{I} - \frac{\mathbf{B}(y_n;\theta_n)\mathbf{u}_n\mathbf{e}^\top}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n}\right)\frac{\left(\partial\mathbf{B}(y_n;\theta_n)/\partial\theta^{(l)}\right)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n} \\ &+ \frac{\left(\partial\mathbf{P}_{\theta_n}^\top/\partial\theta^{(l)}\right)\mathbf{B}(y_n;\theta_n)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_n)\mathbf{u}_n}, \end{aligned}$$

and $\theta^{(l)}$ denotes the l^{th} element of the parameter θ_n, $\mathbf{I}$ denotes the $I \times I$ identity matrix, $\mathbf{e} = [1, \ldots, 1]^\top$, the initial $\omega_0^{(l)}$ is arbitrarily chosen from $\Sigma = \{\omega^{(l)} \in \mathbb{R}^I : e^\top \omega^{(l)} = 0\}$.

At each time step n, the HMM estimator defined by (6), (8), (9), and (10) updates θ_n based on the current sample $y_n = (o_n, r_n, a_n)$, while keeping track of the state estimate $\mathbf{u}_n$, and its partial derivative ω_n.

Now we state the convergence of the estimator.

***Proposition** 2:* Suppose that Assumption 1 and Assumption 2 hold. Then, the following statements hold:

(i) The extended Markov chain $\{s_n, y_n, \mathbf{u}_n, \omega_n\}$ is geometrically ergodic[3].

(ii) For $\theta \in \Theta$, the log-likelihood $l_n(\theta)$ in (1) almost surely converges to $l(\theta)$,

$$l(\theta) = \int_{\mathcal{Y} \times \mathcal{P}(\mathcal{S})} \log[\mathbf{b}^\top(y;\theta)\mathbf{u}] \, \nu(dy, d\mathbf{u}), \tag{11}$$

where $\mathcal{Y} := \mathcal{O} \times \mathbb{R} \times \mathcal{A}$, $\mathcal{P}(\mathcal{S})$ is the set of probability distribution on $\mathcal{S}$, and $\nu(dy, d\mathbf{u})$ is the marginal distribution of ν, which is the invariant distribution of the extended Markov chain.

(iii) The iterate $\{\theta_n\}$ converges almost surely to the invariant set (set of equilibrium points) of the ODE

$$\dot{\theta} = \mathbf{H}(\theta) + \tilde{m} = \Pi_{T_H(\theta)}[\mathbf{H}(\theta)], \quad \theta(0) = \theta_0, \tag{12}$$

where $\mathbf{H}(\theta) = E[\mathbf{S}(y_n, \mathbf{u}_n, \omega_n; \theta)]$, the expectation $E[\cdot]$ is taken with respect to ν, and $\tilde{m}(\cdot)$ is the projection term to keep θ_n in H, $T_H(\theta)$ is the tangent cone of H at θ [14, pp. 343].

***Remark** 1:* The second equation in (12) is due to [16, Appendix E]. Using the definitions of tangent and normal cones [14, pp. 343], we can readily prove that the set of stationary points of (12) is $\{\theta \in H : \Pi_{T_H(\theta)}(\mathbf{H}(\theta)) = 0\} = \{\theta \in H : \mathbf{H}(\theta) \in N_H(\theta)\}$, where $N_H(\theta)$ is the normal cone of H at $\theta \in H$. Note that the set of stationary points is identical to the set of KKT points of the constrained nonlinear programming $\min_{\theta \in H} l(\theta)$.

Proof: We prove that the HMM estimation converges to the invariant set of ODE (12) by verifying the assumptions in [11] for the POMDP with the behavior policy described in Section II. Due to the space limitation, we defer the details of the proof to the online full version [17]. ■

B. Estimating Q-function with the HMM State Predictor

In addition to estimation of the HMM parameters $(\mathbf{P}, \mathbf{R}, \mathbf{O}, \sigma)$, we aim to recursively estimate the optimal action-value function $Q^*(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ using partial state observation.

From Bellman's optimality principle, $Q^*(s,a)$ function is defined as

$$Q^*(s,a) = \sum_{s'} P(s'|s,a) \left(r(s,a) + \gamma \max_{a'} Q^*(s',a') \right), \tag{13}$$

where $P(s'|s,a)$ is the state transition probability, which corresponds to $T_a(s,s')$ in the POMDP model. The standard Q-learning from [18] estimates $Q^*(s,a)$ function using the recursive form:

$$\begin{aligned} &Q_{n+1}(s_n, a_n) \\ &= Q_n(s_n, a_n) + \epsilon_n \left(r_n + \gamma \max_{a'} Q_n(s_{n+1}, a') - Q_n(s_n, a_n) \right). \end{aligned}$$

Since the state s_n is not directly observed in POMDP, the state estimate $\mathbf{u}_n$ in (9) from the HMM estimator is used instead of s_n. Define the estimated state transition $\hat{p}(s_{n-1}, s_n)$ as

$$\begin{aligned} &\hat{p}(s_{n-1}, s_n) \\ &= P(s_{n-1}|y_{n-1}, \mathbf{u}_{n-1}; \theta_{n-1}) P(s_n|y_n, \mathbf{u}_n; \theta_n), \end{aligned} \tag{14}$$

where $P(s_n|y_n, \mathbf{u}_n; \theta_n)$ is calculated using Bayes rule:

$$P(s_n = i|y_n, \mathbf{u}_n; \theta_n) = \frac{b_i(y_n) u_{n,i}}{\sum_j b_j(y_n) u_{n,j}}. \tag{15}$$

Using $\hat{p}(i,j)$ as a surrogate for $P(s'|s,a)$ in (13), a recursive estimator for $Q^*(s,a)$ is proposed as follows:

$$\begin{aligned} &\begin{bmatrix} q_{n+1}(1,a_n) \\ q_{n+1}(2,a_n) \\ \vdots \\ q_{n+1}(I,a_n) \end{bmatrix} = \begin{bmatrix} q_n(1,a_n) \\ q_n(2,a_n) \\ \vdots \\ q_n(I,a_n) \end{bmatrix} + \\ &\epsilon_n \begin{bmatrix} \sum_j^I \hat{p}_n(1,j) \left(r_n + \gamma \max_{a'} q_n(j,a') - q_n(1,a_n) \right) \\ \sum_j^I \hat{p}_n(2,j) \left(r_n + \gamma \max_{a'} q_n(j,a') - q_n(2,a_n) \right) \\ \vdots \\ \sum_j^I \hat{p}_n(I,j) \left(r_n + \gamma \max_{a'} q_n(j,a') - q_n(I,a_n) \right) \end{bmatrix}, \end{aligned} \tag{16}$$

where $q_n(i,a_n) = Q_n(s=i, a=a_n)$. In the following proposition we establish the convergence of (16).

***Proposition** 3:* Suppose that Assumption 1 and Assumption 2 hold. Then the following ODE has a unique globally asymptotically stable equilibrium point:

$$\begin{bmatrix} \dot{q}_{1,a} \\ \dot{q}_{2,a} \\ \vdots \\ \dot{q}_{I,a} \end{bmatrix} = \frac{1}{\bar{u}_a} \begin{bmatrix} \sum_j^I \bar{p}(1,j)(\bar{r} + \gamma \max_{a'} q_{j,a'} - q_{1,a}) \\ \sum_j^I \bar{p}(2,j)(\bar{r} + \gamma \max_{a'} q_{j,a'} - q_{2,a}) \\ \vdots \\ \sum_j^I \bar{p}(I,j)(\bar{r} + \gamma \max_{a'} q_{j,a'} - q_{I,a}) \end{bmatrix}, \quad a \in \mathcal{A},$$

where $\bar{u}_a$ is determined by the expected frequency of the recurrence to the action a (for the detail, see Appendix B in [17]), $\bar{p}(i,j)$ denotes the expectation of $\hat{p}(i,j)$, $\bar{r}$ denotes the expectation of $R(s,a)$ and the expectations are taken with the invariant distribution ν. As a result, the iterate $\{Q_n\}$ of the recursive estimation law in (16) converges in distribution to the unique equilibrium point $\hat{Q}^*$ of the ODE, i.e., the unique solution of the Bellman equation

$$\hat{Q}(s,a) = \sum_{s'} \bar{p}(s,s') \left(\bar{r}(s,a) + \gamma \max_{a'} \hat{Q}(s',a') \right).$$

Proof: The update of Q_n^ϵ is asynchronous, as we update the part of $Q_n(s,a)$ for the current action taken. Result on stochastic approximation from [19] is invoked to prove the convergence. The proof follows from the ergodicity of the underlying Markov chain and the contraction of the operator $HQ = \sum_{s'} \hat{p}(s,s';\theta_L) \left(r(s,a) + \gamma \max_{a'} Q(s',a') \right)$. Due to the space limitation, we defer the details of the proof to the online full version [17]. ■

[3] A Markov chain with transition probability matrix $\mathbf{P}$ is geometrically ergodic, if for finite constants c_{ij} and a $\beta < 1$

$$|(\mathbf{P}^n)_{i,j} - \pi_j| \le c_{ij} \beta^n,$$

where π denotes the stationary distribution.

C. Learning State Transition given Action with the HMM State Predictor

We aim to estimate the expectation of the following indicator function

$$T_{s,a,s'} = E[\mathbb{1}_{\{s_n=s, a_n=j, s_{n+1}=s'\}}], \tag{17}$$

where the expectation E is taken with respect to the stationary distribution corresponding to the true parameter θ^*. Thus, $T_{s,a,s'}$ is the expectation of the counter of the transition s, a, s' divided by the total number of transitions (or the stationary distribution $P(s, a, s')$).

The proposed recursive estimation of $T_{s,a,s'}$ is given by

$$\begin{aligned}\begin{bmatrix} T_{n+1}(1,a_n,1) \\ T_{n+1}(1,a_n,2) \\ \vdots \\ T_{n+1}(I,a_n,I) \end{bmatrix} &= \begin{bmatrix} T_n(1,a_n,1) \\ T_n(1,a_n,2) \\ \vdots \\ T_n(I,a_n,I) \end{bmatrix} \\ &+ \epsilon_n \begin{bmatrix} \hat{p}_n(1,1)(1-T_n(1,a_n,1)) \\ \hat{p}_n(1,2)(1-T_n(1,a_n,2)) \\ \vdots \\ \hat{p}_n(I,I)(1-T_n(I,a_n,I)) \end{bmatrix}.\end{aligned} \tag{18}$$

We note that the estimation in (18) uses $\hat{p}(s, s')$ as a surrogate for $P(s'|s, a)$ in (13). The ODE corresponding to (18) is

$$\begin{bmatrix} \dot{T}_{1,a,1} \\ \dot{T}_{1,a,2} \\ \vdots \\ \dot{T}_{I,a,I} \end{bmatrix} = \frac{1}{\bar{u}_a} \begin{bmatrix} \bar{p}(1,a,1)(1-T_{1,a,1}) \\ \bar{p}(1,a,2)(1-T_{1,a,2}) \\ \vdots \\ \bar{p}(I,a,I)(1-T_{I,a,I}) \end{bmatrix}, \quad a \in \mathcal{A}.$$

Following the same procedure in the proof of *Proposition 3*, we can show that $t_n(s, a, s')$ converges to $\bar{p}(s, a, s')$, where $\bar{p}(s, a, s')$ denotes the marginal distribution of the transition from s to s' after taking a with respect to the invariant distribution of the entire process. Since we estimate the joint distribution, the conditional distribution $T_a(s, s')$ can be calculated by dividing the joint probabilities with marginal probabilities.

IV. A Numerical Example

In this simulation, we implement the HMM Q-learning for a finite state POMDP example, where 4 hidden states are observed through 2 observations with the discount factor $\gamma = 0.95$ as specified below:

$$\mathbf{T} = \left[\begin{bmatrix} .6 & .2 & .1 & .1 \\ .2 & .1 & .6 & .1 \\ .1 & .1 & .1 & .7 \\ .4 & .1 & .1 & .4 \end{bmatrix}, \begin{bmatrix} .1 & .2 & .2 & .5 \\ .1 & .6 & .1 & .2 \\ .1 & .2 & .6 & .1 \\ .1 & .1 & .2 & .6 \end{bmatrix} \right],$$

$$\mathbf{O} = \begin{bmatrix} .95 & .05 \\ .95 & .05 \\ .05 & .95 \\ .05 & .95 \end{bmatrix}, \ \mathbf{R} = \begin{bmatrix} 0 & 0. & -20. & +20. \\ 0 & 0. & +20. & -20. \end{bmatrix}, \ \sigma = 1.$$

The following behavior policy $\mu(o)$ is used to estimate the HMM, the transition model, and the Q-function

$$\mu = \begin{bmatrix} .6 & .4 \\ .3 & .7 \end{bmatrix}, \quad \mu_{i,j} = P(a = j|o = i).$$

The diminishing step size is chosen as $\epsilon_n = n^{-0.4}$ for $n \geq 1$.

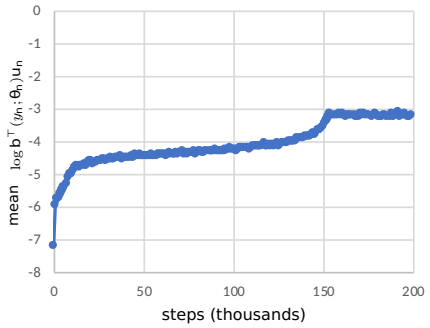

Fig. 2: The mean of $\log \mathbf{b}^\top(y_n; \theta_n)\mathbf{u}_n$.

A. Estimation of the HMM and Q-function

Figure 2 shows that the mean of the sample conditional log-likelihood density $\log \mathbf{b}^\top(y_n; \theta_n)\mathbf{u}_n$ increases.

To validate the estimation of the Q-function in (16), we run three estimations of Q-function in parallel: (i) Q-learning [18] with full state observation s, (ii) Q-learning with partial observation o, (iii) HMM Q-learning. Figure 3 shows $\max_{s,a} Q_n(s, a)$ for all three algorithms.

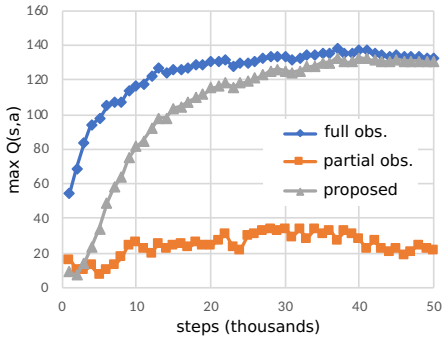

Fig. 3: $\max_{s,a} Q_n(s, a)$ is greater with full observation than partial observation.

After 200,000 steps, the iterates of Q_n^{full}, Q_n^{partial} and Q_n^{hmm} at $n = 2 \times 10^5$ are as follows:

$$Q_n^{\text{full}} = \begin{bmatrix} 107.4 & 103.4 & 99.3 & 133.8 \\ 114.7 & 107.6 & 102.4 & 98.0 \end{bmatrix}^\top,$$

$$Q_n^{\text{partial}} = \begin{bmatrix} 20.1 & 21.6 \\ 18.9 & 9.1 \end{bmatrix}^\top,$$

$$Q_n^{\text{hmm}} = \begin{bmatrix} 133.0 & 106.0 & 105.9 & 99.1 \\ 98.1 & 111.2 & 111.7 & 105.4 \end{bmatrix}^\top,$$

where the (i, j) elements of the Q matrices are the estimates of the Q-function value, when $a = i, s = j$. Similar to the other HMM estimations (from unsupervised learning task), the labels of the inferred hidden state do not match the labels assigned to the true states. Permuting the state indices $\{1, 2, 3, 4\}$ to $(2, 3, 4, 1)$ in order to have better matching between the estimated and true Q-function, we compare the

estimated Q-function as follows:

$$Q_n^{\text{permuted}} = \begin{bmatrix} 106.0 & 105.9 & 99.1 & 133.0 \\ 111.2 & 111.7 & 105.4 & 98.1 \end{bmatrix}^\top,$$

$$Q_n^{\text{full}} = \begin{bmatrix} 107.4 & 103.4 & 99.3 & 133.8 \\ 114.7 & 107.6 & 102.4 & 98.0 \end{bmatrix}^\top.$$

This permutation is consistent with the estimated observation $\mathbf{O}(\theta_n)$ as below:

$$\mathbf{O}(\theta_n) = \begin{bmatrix} .066 & .934 \\ .943 & .057 \\ .947 & .053 \\ .052 & .948 \end{bmatrix}, \quad \mathbf{O}(\theta^*) = \begin{bmatrix} .950 & .050 \\ .950 & .050 \\ .050 & .950 \\ .050 & .950 \end{bmatrix}.$$

B. Dynamic Policy with Partial Observations

After a certain stopping criterion is satisfied, we fix the parameter. The fixed POMDP parameters $(\mathbf{T}_{\theta_l}, \mathbf{O}_{\theta_l}, \mathbf{R}_{\theta_l}, \sigma_{\theta_l})$ are used in the following Bayesian state belief filter

$$\mathbf{u}_{n+1} = \frac{\mathbf{T}_{\theta_l}^\top(a_n)\mathbf{B}(y_n;\theta_l)\mathbf{u}_n}{\mathbf{b}^\top(y_n;\theta_l)\mathbf{u}_n}. \tag{19}$$

The action a^* is chosen based on the expectation of the Q-function on the state belief distribution and the current observation o_n

$$a^* = \arg\max_a \sum_i^I Q_{\theta_l}(s=i,a)P(s_n=i|o_n,\mathbf{u}_n;\theta_l). \tag{20}$$

We tested the dynamic policy consisting of (19) and (20) at every thousand steps of the parameter estimation. Figure 4 shows that the proposed HMM Q-learning performs better than the Q-learning with partial observation.

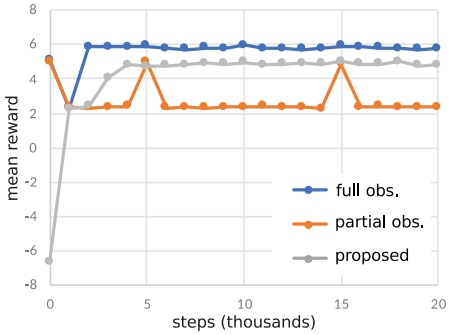

Fig. 4: mean rewards from Q-learning with full observation, Q-learning with partial observation, and the proposed HMM Q-learning.

V. Conclusion

We presented a model-based approach to the problem of reinforcement learning with incomplete observation. Since the controlled POMDP is an HMM, we invoked results from Hidden Markov Model (HMM) estimation. Based on the convergence of the HMM estimator, the optimal action-value function $Q^*(s,a)$ is learned despite the hidden states. The proposed algorithm is recursive, i.e. only the current sample is used so that there is no need for replay buffer, in contrast to the other algorithms for POMDP [20], [21].

We proved the convergence of the recursive estimator using the ergodicity of the underlying Markov chain for the HMM estimation [11], [12] and presented a numerical example where the simulation shows the convergent behavior of the recursive estimator.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., "Mastering the game of Go without human knowledge," Nature, vol. 550, no. 7676, p. 354, 2017.

[3] S. P. Singh, T. Jaakkola, and M. I. Jordan, "Learning without state-estimation in partially observable Markovian decision processes," in Machine Learning Proceedings 1994. Elsevier, 1994, pp. 284–292.

[4] W. S. Lovejoy, "A survey of algorithmic methods for partially observed Markov decision processes," Annals of Operations Research, vol. 28, no. 1, pp. 47–65, 1991.

[5] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in Machine Learning Proceedings 1995. Elsevier, 1995, pp. 362–370.

[6] H. Yu and D. P. Bertsekas, "Discretized approximations for POMDP with average cost," in Proceedings of the 20th conference on Uncertainty in artificial intelligence. AUAI Press, 2004, pp. 619–627.

[7] L. Chrisman, "Reinforcement learning with perceptual aliasing: The perceptual distinctions approach," in AAAI, vol. 1992, 1992, pp. 183–188.

[8] S. Ross, B. Chaib-draa, and J. Pineau, "Bayes-adaptive POMDPs," in Advances in Neural Information Processing Systems, 2008, pp. 1225–1232.

[9] P. Karkus, D. Hsu, and W. S. Lee, "QMDP-Net: Deep learning for planning under partial observability," in Advances in Neural Information Processing Systems, 2017, pp. 4697–4707.

[10] Z. D. Guo, S. Doroudi, and E. Brunskill, "A PAC RL algorithm for episodic POMDPs," in Artificial Intelligence and Statistics, 2016, pp. 510–518.

[11] V. Krishnamurthy and G. G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," IEEE Transactions on Information Theory, vol. 48, no. 2, pp. 458–476, 2002.

[12] F. LeGland and L. Mevel, "Recursive estimation in hidden Markov models," in Decision and Control, 1997., Proceedings of the 36th IEEE Conference on, vol. 4. IEEE, 1997, pp. 3468–3473.

[13] J. R. Norris, Markov chains. Cambridge university press, 1998, no. 2.

[14] D. P. Bertsekas, Nonlinear programming. Athena scientific Belmont, 1999.

[15] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," The annals of mathematical statistics, vol. 41, no. 1, pp. 164–171, 1970.

[16] S. Bhatnagar, H. Prasad, and L. Prashanth, Stochastic recursive algorithms for optimization: simultaneous perturbation methods. Springer, 2012, vol. 434.

[17] H.-J. Yoon, D. Lee, and N. Hovakimyan, "Hidden Markov model estimation-based Q-learning for partially observable Markov decision process," arXiv preprint arXiv:1809.06401, 2018.

[18] C. J. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, no. 3-4, pp. 279–292, 1992.

[19] H. Kushner and G. G. Yin, Stochastic approximation and recursive algorithms and applications. Springer Science & Business Media, 2003, vol. 35.

[20] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," CoRR, abs/1507.06527, 2015.

[21] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," arXiv preprint arXiv:1512.04455, 2015.