Approximate information state for partially observed systems

Jayakumar Subramanian
Electrical and Computer Engineering
McGill University, Montreal, QC H2T2A7
jayakumar.subramanian@mail.mcgill.ca

Aditya Mahajan
Electrical and Computer Engineering
McGill University, Montreal, QC H2T2A7
aditya.mahajan@mcgill.ca

Abstract

The standard approach for modeling partially observed systems is to model them as partially observable Markov decision processes (POMDPs) and obtain a dynamic program in terms of a belief state. The belief state formulation works well for planning but is not ideal for learning because the belief state depends on the model and, as such, is not observable when the model is unknown.

In this paper, we present an alternative notion of an information state for obtaining a dynamic program in partially observed models. In particular, an information state is a sufficient statistic for the current reward which evolves in a controlled Markov manner. We show that such an information state leads to a dynamic programming decomposition. Then we present a notion of an approximate information state and present an approximate dynamic program based on the approximate information state. Approximate information state is defined in terms of properties that can be estimated using sampled trajectories. Therefore, they provide a constructive method for reinforcement learning in partially observed systems. We present one such construction and show that it performs better than the state of the art for three benchmark models.

Keywords: Partially observable Markov decision processes; reinforcement learning; planning; approximate information state.

1 Introduction

The theory of Markov decision processes focuses primarily on systems with full state observation. When systems with partial state observations are considered, they are converted to systems with full state observations by considering the belief state (which is the posterior belief on the state of the system given the history of observations and actions). Although this leads to an explosion in the size of the state space, the resulting value function has a nice property—it is piecewise linear and convex in the belief state [14]—which is exploited to develop efficient algorithms to compute the optimal policy [9, 13]. Thus, for planning, there is little value in studying alternative characterizations of partially observed models.

However, the belief state formulation is not as nice a fit for learning. Part of the difficulty is that the construction of the belief state depends on the system model. So, when the system model is unknown, the belief state cannot be constructed using the observations. Therefore, critic based methods are not directly applicable. There are some results that circumvent this difficulty [3, 7, 10]. However, many of the recent results suggest that using RNNs (Recurrent Neural Networks [12]) or LSTMs (Long Short Term Memories [8]) for modeling the policy function (actor) and/or the action-value function (critic) works for reinforcement learning in partially observed systems [1, 2, 5, 6, 16, 17]. In this paper, we present a rigorous theory for planning and learning in partially observed models using the notions of information state and approximate information state. We then present numerical experiments that show that the approximate information state based works well on benchmark models.

2 Model

A general system with partial observations may be represented using the following stochastic input-output model. Consider a system that takes two inputs: a control input $U_t \in \mathcal{U}$ and a stochastic input $W_t \in W$ and generates two outputs: an observation $Y_t \in \mathcal{Y}$ and a real-valued reward R_t. The spaces $\mathcal{W}, \mathcal{U}$, and $\mathcal{Y}$ are Banach spaces and the stochastic inputs $(W_1, \dots, W_T)$ are independent random variables defined on a common probability space.

Formally, we assume that there are observation functions $\{f_t\}_{t=1}^T$ and reward functions $\{r_t\}_{t=1}^T$ such that $Y_t = f_t(Y_{1:t}, U_{1:t-1}, W_t)$ and $R_t = r_t(Y_{1:t}, U_{1:t}, W_t)$. An agent observes the history $H_t = (Y_{1:t}, U_{1:t-1})$ of observations and control inputs until time t and chooses the control input $U_t = \pi_t(H_t)$ according to some history dependent policy $\pi := \{\pi_t\}_{t=1}^T$. The performance of policy π is given by

$$J(\pi) = \mathbb{E}^\pi\left[\sum_{t=1}^T R_t\right]. \tag{1}$$

The objective of the agent is to choose a policy π to maximize the expected total reward $J(\pi)$.

A dynamic programming decomposition. Recursively define the following *value functions*. $V_{T+1}(h_{T+1}) := 0$ and for $t \in \{T, \dots, 1\}$:

$$Q_t(h_t, u_t) = \mathbb{E}[R_t + V_{t+1}(H_{t+1}) \mid H_t = h_t, U_t = u_t] \quad \text{and} \quad V_t(h_t) = \max_{u_t \in \mathcal{U}} Q_t(h_t, u_t). \tag{2}$$

Theorem 1 *A policy $\pi = (\pi_1, \dots, \pi_T)$ is optimal if and only if it satisfies $\pi_t(h_t) \in \arg\max_{u_t \in \mathcal{U}} Q_t(h_t, u_t)$.*

The above dynamic program uses the history of observations and actions as state and as such a dynamic program is not efficient for computing the optimal policy but it will serve as a reference for the rest of the analysis.

Information state and a simplified dynamic program. Let $\mathcal{F}_t = \sigma(H_t)$ denote the filtration generated by the history of observations and control actions.

Definition 1 An information state $\{Z_t\}_{t\geq 1}$, $Z_t \in \mathcal{Z}$, is an $\mathcal{F}_t$ adapted process (therefore, there exist functions $\{\vartheta_t\}_{t=1}^T$ such that $Z_t = \vartheta_t(H_t)$) that satisfies the following properties:

(P1) **Sufficient for performance evaluation**, i.e., $\mathbb{E}[R_t \mid H_t = h_t, U_t = u_t] = \mathbb{E}[R_t \mid Z_t = \vartheta_t(h_t), U_t = u_t]$.

(P2) **Sufficient to predict itself**, i.e., $\mathbb{P}(Z_{t+1} = z_{t+1} | H_t = h_t, U_t = u_t) = \mathbb{P}(Z_{t+1} = z_{t+1} | Z_t = z_t, U_t = u_t)$, for all z_{t+1}.

There is no restriction on the space $\mathcal{Z}$, although an information state is useful only when the space $\mathcal{Z}$ is "small" in an appropriate sense. We have assumed that the space $\mathcal{Z}$ is time-homogeneous for convenience. In some situations, it may be more convenient to construct an information state which takes values in spaces that are changing with time.

For some models, instead of (P2), it is easier to verify the following stronger conditions:

(P2a) **Evolves in a state-like manner**, i.e., there exist measurable functions $\{\varphi_t\}_{t=1}^T$ such that $Z_{t+1} = \varphi_t(Z_t, Y_{t+1}, U_t)$.

(P2b) Is sufficient for predicting future observations, i.e., for any y_{t+1}

$$\mathbb{P}(Y_{t+1} = y_{t+1} \mid H_t = h_t, U_t = u_t) = \mathbb{P}(Y_{t+1} = y_{t+1} \mid Z_t = \vartheta_t(h_t), U_t = u_t).$$

Proposition 1 *(P2a) and (P2b) imply (P2).*

Note that $Z_t = H_t$ is always an information state, so an information state always exists. It is straight-forward to show that if we construct a state space model for the above input-output model, then the belief on the state given the history of observations and controls is an information state. Below we present an example of a non-trivial information state that is much simpler than the belief state.

Example 1 (Machine Maintenance) Consider a machine which can be in one of n ordered states where the first state is the best and the last state is the worst. The production cost increases with the state of the machine. The state evolves in a Markovian manner. At each time, an agent has the option to either run the machine or stop and inspect it for a cost. After inspection, s/he may either repair it (at a cost that depends on the state) or replace it (at a fixed cost). The objective is to identify a maintenance policy to determine to minimize the cost of production, inspection, repair, and replacement.

Let τ denote the time of last inspection and S_τ denote the state of the machine after inspection, repair, or replacement. Then, it can be shown that $(S_\tau, t - \tau)$ is an information state for the system.

The main feature of an information state is that one can always write a dynamic program based on an information state.

Theorem 2 *Let $\{Z_t\}_{t=1}^T$ be an information state. Recursively define value functions $\{\tilde{V}_t\}_{t=1}^{T+1}$, where $\tilde{V}_t \colon Z_t \mapsto \mathbb{R}$ as follows: $\tilde{V}_{T+1}(z_{T+1}) = 0$ and for $t \in \{T, \dots, 1\}$:*

$$\tilde{Q}_t(z_t, u_t) = \mathbb{E}[R_t + \tilde{V}_{t+1}(Z_{t+1}) \mid Z_t = z_t, U_t = u_t] \quad \text{and} \quad \tilde{V}_t(z_t) = \max_{u_t \in \mathcal{U}} Q_t(z_t, u_t). \tag{3}$$

Then, $Q_t(h_t, u_t) = \tilde{Q}_t(\vartheta_t(h_t), u_t)$ and $V_t(h_t) = \tilde{V}_t(\vartheta_t(h_t))$.

Remark 1 In light of Theorem 2, an information state may be viewed as a generalization of the traditional notion of state [11, 18]. Traditionally, the state of an input-output system is sufficient for input-output mapping. In contrast, the information state is sufficient for dynamic programming.

The notion of information state is also related to sufficient statistics for optimal control [15]. However, in contrast to [15], we do not assume a state space model for the underlying system so it is easier to develop reinforcement learning algorithms using our notion of an information state.

Coming back to Example 1, Theorem 2 shows that we can write a dynamic program for that model using the information state $(S_\tau, t - \tau)$, which takes values in a countable set. This countable state dynamic program is considerably simpler than the standard belief state dynamic program typically used for that model. Another feature of the information state formulation is that the information state $(S_\tau, t - \tau)$ does not depend on the transition probability of the state of the machine or the cost of inspection or repair. Thus, if these model parameters were unknown, we can use a standard reinforcement learning algorithm to find an optimal policy which maps $(S_\tau, t - \tau)$ to current action.

Given these benefits of a good information state, it is natural to consider a data-driven approach to identify an information state. An information state identified from data will not be exact and it is important to understand what is the loss in performance when using an approximate information state. In the next section, we present a notion of approximate information state and bound the approximation error.

3 Approximate information state (AIS)

Roughly speaking, a compression of the history is an approximate information state if it approximately satisfies (P1) and (P2). This intuition can be made precise as follows.

Definition 2 Given positive numbers ε and δ, an (ε, δ)-approximate information state $\{\hat{Z}_t\}_{t=1}^T$, where $\hat{Z}_t$ takes values in a in a Polish metric space $(\hat{\mathcal{Z}}, d)$, is an $\mathcal{F}_t$ adapted process (therefore, there exist functions $\{\widehat{\vartheta}_t\}_{t=1}^T$ such that $\hat{Z}_t = \widehat{\vartheta}_t(H_t)$) that satisfies the following properties:

(AP1) Sufficient for approx. performance evaluation, i.e., $\big|\mathbb{E}[R_t \mid H_t = h_t, U_t = u_t] - \mathbb{E}[R_t \mid \hat{Z}_t = \widehat{\vartheta}_t(h_t), U_t = u_t]\big| \le \varepsilon$.

(AP2) Sufficient to predict itself approximately. For any Borel subset A of $\hat{\mathcal{Z}}$ define, $\mu_t(A) = \mathbb{P}(\hat{Z}_{t+1} \in A \mid H_t = h_t, U_t = u_t)$ and $\nu_t(A) = \mathbb{P}(\hat{Z}_{t+1} \in A \mid \hat{Z}_t = \widehat{\vartheta}_t(h_t), U_t = u_t)$. Then, $\mathcal{K}_d(\mu_t, \nu_t) \le \delta$, where $\mathcal{K}_d(\cdot, \cdot)$ denotes the Wasserstein or Kantorovich-Rubinstein distance[1] between two distributions.

[1]Let $(\mathcal{X}, d)$ be a Polish metric space. For any two probability measures μ, ν on $\mathcal{X}$, the Wasserstein distance between μ and ν is: $\mathcal{K}_d(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X}} d(x, y)^p d\pi(x, y)$ where Π represents the product space of the two distributions.

Our main result is that one can write an approximate dynamic program based on an approximate information state.

Theorem 3 *Let $\{\hat{Z}_t\}_{t=1}^T$ be an (ε, δ)-approximate information state. Recursively define value functions $\{\widehat{V}_t\}_{t=1}^{T+1}$, where $\widehat{V}_t \colon \hat{Z}_t \mapsto \mathbb{R}$ as follows: $\widehat{V}_{T+1}(\hat{z}_{T+1}) = 0$ and for $t \in \{T, \dots, 1\}$:*

$$\widehat{Q}_t(\hat{z}_t, u_t) = \mathbb{E}[R_t + \widehat{V}_{t+1}(\hat{Z}_{t+1}) \mid \widehat{Z}_t = \hat{z}_t, U_t = u_t] \quad \text{and} \quad \widehat{V}_t(\hat{z}_t) = \max_{u_t \in \mathcal{U}} \widehat{Q}_t(\hat{z}_t, u_t).$$

Suppose $\widehat{V}_t$ is Lipschitz continuous with Lipschitz constant L_V. Then, we have the following:

$$|Q_t(h_t, u_t) - \widehat{Q}_t(\widehat{\vartheta}_t(h_t), u_t)| \le (T-t)(\varepsilon + L_V \delta) + \varepsilon \quad \text{and} \quad |V_t(h_t) = \widehat{V}_t(\widehat{\vartheta}_t(h_t))| \le (T-t)(\varepsilon + L_V \delta) + \varepsilon.$$

Based on Prop. 1, we provide an alternative characterization of an approximate information state. We can replace (AP2) with the following stronger conditions:

(AP2a) **Evolves in a state-like manner**, i.e., there exist measurable functions $\{\hat{\varphi}_t\}_{t=1}^T$ such that $\hat{Z}_{t+1} = \hat{\varphi}_t(\hat{Z}_t, Y_{t+1}, U_t)$. Moreover, these functions are Lipschitz in Y with Lipschitz constant L_U.

(AP2b) **Is sufficient for predicting future observations approximately**. For any Borel subset A of $\mathcal{Y}$ define, $\mu_t(A) = \mathbb{P}(Y_{t+1} \in A \mid H_t = h_t, U_t = u_t)$ and $\nu_t(A) = \mathbb{P}(Y_{t+1} \in A \mid \hat{Z}_t = \widehat{\vartheta}_t(h_t), U_t = u_t)$. Then, $\mathcal{K}(\mu_t, \nu_t) \le \delta$,

Proposition 2 *If (AP2) is replaced by (AP2a) and (AP2b), the result of Theorem 3 holds with L_V replaced by $L_U L_V$.*

Corollary 1 *Suppose $\{Z_t\}_{t=1}^T$ is an information state and $\{\hat{Z}_t\}_{t=1}^T$ is an (ε, δ)-approximate information state. Then for any realization h_t of H_t, we have the following:*

$$|Q_t(\vartheta_t(h_t), u_t) - \widehat{Q}_t(\widehat{\vartheta}_t(h_t), u_t)| \le (T-t)(\varepsilon + L_V \delta) + \varepsilon \quad \text{and} \quad |V_t(\vartheta_t(h_t)) - \widehat{V}_t(\widehat{\vartheta}_t(h_t))| \le (T-t)(\varepsilon + L_V \delta) + \varepsilon.$$

4 Reinforcement learning using approximate information state

In this section, we use an approximate information state to design reinforcement learning algorithms for infinite horizon POMDPs. We split our approach into two steps—a data-driven approach to construct an approximate information state and reinforcement learning using this approximate information state.

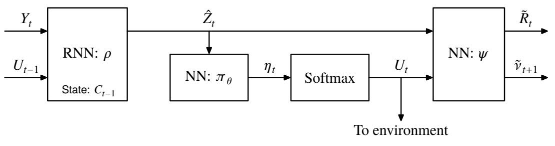
Constructing an approximate information state. The definition of approximate information state suggests two ways to construct an information state from data: either use $\hat{\vartheta}(h_t)$ to determine an approximate information state that satisfies conditions (AP1) and (AP2) or conditions (AP1), (AP2a), and (AP2b). We present the second approach here.

We use two function approximators: (i) A recurrent neural network (RNN) or its refinements such as LSTM or GRU with state C_{t-1}, inputs (Y_t, U_{t-1}) and output $\hat{Z}_t$. We denote this function approximator by ρ. (ii) A feed forward network as in the previous case, except that $\tilde{\nu}_{t+1}$ is the prediction of ν_{t+1}, the distribution of the next approximate information state $\hat{Z}_{t+1}$. We denote this function approximator as ψ.

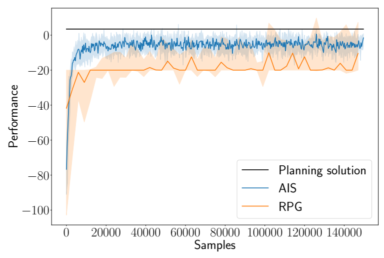
To minimize ε and δ, we train the networks ρ and ψ using the loss function $\mathcal{L}_{\rho,\psi} = \lambda \mathcal{L}_R + (1-\lambda)\mathcal{L}_\nu$ where $\mathcal{L}_R = \frac{1}{B}\sum_{t=1}^{B} \texttt{smoothL1}(\tilde{R}_t - R_t)$, (where B is the batch size and `smoothL1` is the standard smooth approximation for L1 loss) and $\mathcal{L}_\nu = -\sum_{t=1}^{B-1} \log(\tilde{\nu}_{t+1}(Y_{t+1}))$, which is the negative log likelihood loss for $\tilde{\nu}_t$ and thus approximates the KL-divergence between μ_t and ν_t. We use the KL-divergence as a surrogate for the Wasserstein distance because: (i) Wasserstein distance is computationally expensive to compute; and (ii) KL-divergence upper bounds the total variation (due to Pinsker's inequality), which in turn upper bounds Wasserstein distance for metric spaces with bounded diameter.

Reinforcement learning. Let $\pi_\theta : \hat{Z}_t \mapsto \Delta(U_t)$ be a parametrized stochastic policy, where the parameters θ lie in a closed convex set Θ. For example, π_θ could be a feed forward neural network with input $\hat{Z}_t$ and output to be a $|U_t|$ dimensional vector η, where: $\pi_\theta(u|\hat{z}) = \exp(\tau\eta_u)/\sum_{w\in\mathcal{U}} \exp(\tau\eta_w)$, where τ is a hyperparameter. In such a policy, θ corresponds to the weights of the network. The basic idea behind policy based reinforcement learning is to get sample path based estimates of the performance gradient $\nabla_\theta J$, which is then used as a gradient loss function for updating the parameters θ using stochastic gradient descent.

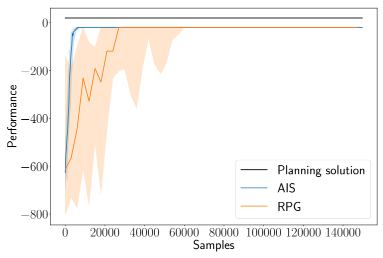
An architecture for combining the construction of the approximate information state with reinforcement learning is shown on the right. In this architecture, we train the networks (ρ, ϕ) and π_θ in parallel using a two time-scale algorithm. In particular, by a slight abuse of notation, let ρ and ψ denote the weights of the corresponding networks. Then,



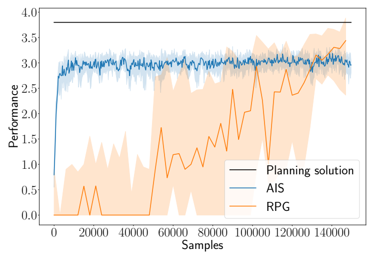
$$\begin{bmatrix}\rho_{k+1}\\ \psi_{k+1}\end{bmatrix} = \begin{bmatrix}\rho_k\\ \psi_k\end{bmatrix} + a_k \nabla_{\rho,\psi}\mathcal{L}_{\rho,\psi} \text{ and } \theta_{k+1} = \theta_k + b_k \nabla_\theta J(\pi_{\theta_k}),$$



(a) Voicemail problem



(b) Tiger problem



(c) 4×4 grid problem

Figure 1: Performance versus samples for all examples. The solid line shows the median value and the shaded region shows the region between the first and third quartiles over 25 runs.

where the learning rates $\{a_k\}_{k\geq 1}$ and $\{b_k\}_{k\geq 1}$ satisfy the standard two time-scale stochastic approximation conditions [4].

The results of the experiment for three small dimensional POMDP benchmarks—voicemail, tiger, and 4×4 grid—are shown in Fig. 1.

References

[1] A. Baisero and C. Amato. Learning internal state models in partially observable environments;. *Reinforcement Learning under Partial Observability, NeurIPS Workshop*, 2018.

[2] B. Bakker. Reinforcement learning with long short-term memory. In *NIPS*, 2002.

[3] J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *JAIR*, 15:319–350, 2001.

[4] V. S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[5] M. Hausknecht and P. Stone. Deep recurrent Q-learning for partially observable MDPs. In *2015 AAAI Fall Symposium Series*, 2015.

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

[7] A. Hefny, Z. Marinho, W. Sun, S. Srinivasa, and G. Gordon. Recurrent predictive state policy networks. *arXiv:1803.01489*, 2018.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[10] M. L. Littman, R. S. Sutton, and S. P. Singh. Predictive representations of state. In *NIPS*, 2002.

[11] A. Nerode. Linear automaton transformations. *Proceedings of American Mathematical Society*, 9:541–544, 1958.

[12] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

[13] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *AAMAS*, 2013.

[14] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[15] C. Striebel. Sufficient statistics in the optimal control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12:576–592, 1965.

[16] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber. Solving deep memory POMDPs with recurrent policy gradients. In *International Conference on Artificial Neural Networks*, 2007.

[17] D. Wierstra, A. Förster, J. Peters, and J. Schmidhuber. Recurrent policy gradients. *Logic Journal of the IGPL*, 18(5):620–634, 2010.

[18] H. S. Witsenhausen. Some remarks on the concept of state. In Y. C. Ho and S. K. Mitter, editors, *Directions in Large-Scale Systems*, pages 69–75. Plenum, 1976.