Towards Using Fully Observable Policies for POMDPs

András Attila Sulyok  
Faculty of Information Technology and Bionics  
Pázmány Péter Catholic University  
Budapest, Hungary  
sulyok.andras.attila@itk.ppke.hu

Kristóf Karacs  
Faculty of Information Technology and Bionics  
Pázmány Péter Catholic University  
Budapest, Hungary  
karacs@itk.ppke.hu

Abstract—Partially Observable Markov Decision Process (POMDP) is a framework applicable to many real world problems. In this work, we propose an approach to solve POMDPs with multimodal belief by relying on a policy that solves the fully observable version. By defining a new, mixture value function based on the value function from the fully observable variant, we can use the corresponding greedy policy to solve the POMDP itself. We develop the mathematical framework necessary for discussion, and introduce a benchmark built on the task of Reconnaissance Blind TicTacToe. On this benchmark, we show that our policy outperforms policies ignoring the existence of multiple modes.

Index Terms—reinforcement learning, pomdp, value function

I. INTRODUCTION

Markov Decision Process (MDP) is a general framework that can be used in many practical applications where there is an agent that is in interaction with its environment, and there is a reward function that the agent needs to maximize [1]. Numerous algorithms have been proposed and are in use to effectively solve MDPs (for example, [2]–[4]).

On the other hand, MDPs assume that the agent knows the state the environment is in at any time, i.e., all information to plan ahead is available. Partially Observable Markov Decision Process (POMDP, [5]) is a modification to the MDP theory that includes partial visibility: instead of observing the state directly, it is observed indirectly via an observation function. This framework is applicable in many areas from autonomous driving to healthcare [6] or education [7].

One way to deal with the uncertainty introduced by partial observability is to keep track of the posterior probability of the current state given the observation and action history, i.e., the belief [8]. In this work, we focus on a specific class of POMDPs, inspired by the recent challenge of Reconnaissance Blind Chess (RBC, [9]–[11]), in which the belief is a multimodal distribution, that is, it is not enough to keep track of only one state, which is presumed to be a corrupted version of the true underlying state of the environment, but instead there are multiple distinct states with too high probability to be simply ignored. We assume that the belief can be calculated.

We also assume that we have access to a (reasonably good) policy that solves the fully observable case, that is, the same environment but with the observation function being the identity function. For many applications, this is not an unreasonable assumption, for example, one can build a simulator for the agent and do training there (see, for example [12]).

Based on these, we propose a mixture policy to solve POMDPs that uses the policy for the fully observable case weighted by the belief. We introduce the necessary mathematical framework for this, and show its performance on a benchmark that is a variation of RBC called Reconnaissance Blind TicTacToe.

A. Related work

The traditional approach to solving POMDPs is to transform the problem into belief-MDPs [8], [13], [14], which are MDPs that use the beliefs of the original problem as their state space. Since this means an exponentially large state space, various estimation techniques were developed [13]; however, these are hard to generalize for continuous state spaces.

There are several attempts to solve POMDPs using deep reinforcement learning techniques, by observation aggregation [2] or recurrent models [15], [16], however, these were not specifically designed for tasks involving multimodal belief. This can result in undesired linear combinations of modes, i.e., instead of keeping track of probable states, the model might keep track of states of lower probability instead, corresponding to combinations of two or more modes.

There are also works proposing to solve a POMDP using a policy for the fully observable case, especially for the RBC challenge [17]–[20], however, these do not give a clear mathematical description and do not discuss their solutions in terms of general POMDPs. They also focus more on the speciality of RBC that the location of the observation is directly controlled by the agent and on how to limit the number of probable states.

II. METHODS

A. POMDP background

In general, POMDPs [5] are defined by the tuple $\langle S, A, \mathcal{O}, \mathcal{P}, O, R, \gamma \rangle$, where $S$ denotes the (unobservable) states, $A$ the set of actions, $\mathcal{O}$ the set of observations and
R the reward function. We denote the state, action, reward and observation at timestep t as the random variables $S_t \in \mathcal{S}$, $A_t \in \mathcal{A}$, $R_t \in \mathbb{R}$ and $O_t \in \mathcal{O}$, respectively. The dynamics is governed by the transition distribution $P(s', r \mid s, a) = \Pr(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a)$ (for all), and the observation function $O(o \mid s) = \Pr(O_t = o \mid S_t = s)$.

To simplify notation, we denote observations up to timestep $t$ as $o_{\leq t} = (o_0, \ldots, o_t)$, and actions before timestep $t$ as $a_{< t} = (a_0, \ldots, a_{t-1})$.

A policy is a distribution over actions: $\pi(a \mid o_{\leq t}, a_{< t}) = \Pr(A_t = a \mid o_{\leq t}, a_{< t})$ given the history of past observations and actions $o_0, a_0, \ldots, a_t$. The goal of a POMDP is to find a policy so that the expected cumulative reward is maximal:

$$\arg \max_{\pi} \mathbb{E}_{(s_t, A_t, R_{t+1}) \sim P, \pi} \sum_t \gamma^t R_{t+1}.$$  

We denote the belief over possible states as $\beta_t = \Pr(S_t \mid o_{\leq t}, a_{< t})$.

Following the POMDP literature [8], we define the belief-MDP of a POMDP as the tuple $\langle B, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$, where $B = \Delta^S$ is the state space of beliefs, $R$ is the reward function:

$$R(\beta) = \mathbb{E}_{s \sim \beta} R(s).$$  

(1)

The dynamics of the belief-MDP is governed by a transition function of the beliefs: after issuing an action, the agent observes a new observation, and transitions into a new belief-state. The distribution of the next observation $o_{t+1}$, given current belief $\beta_t$ and action $a_t$ is:

$$p(o_{t+1} \mid \beta_t, a_t) = \sum_{s_t} \beta_t(s_t) \sum_{s_{t+1}} \mathcal{P}(s_{t+1} \mid s_t, a_t) O(o_{t+1} \mid s_{t+1})$$

(2a)

$$= \mathbb{E}_{s_t \sim \beta_t, s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)} O(o_{t+1} \mid s_{t+1}).$$

(2b)

After the observation, the belief-transition is deterministic:

$$\beta_{t+1}(\beta_t, a_t, o_{t+1}) = p(s_t+1 \mid \beta_t, a_t, o_{t+1})$$

$$= \mathbb{E}_{s_t \sim \beta_t} p(s_{t+1} \mid s_t, a_t, o_{t+1})$$

(3a)

$$= \mathbb{E}_{s_t \sim \beta_t} \left\{ \frac{p(o_{t+1} \mid s_{t+1}, s_t, a_t) p(s_{t+1} \mid s_t, a_t)}{p(o_{t+1} \mid s_t, a_t)} \right\}$$

(3c)

We can define the value function for POMDPs:

$$Q^\pi(\beta, a) = \mathbb{E}_{\beta_t \sim \mathcal{P}, a_t \sim \pi} \left\{ \sum_{t=0}^T R(\beta_t) \right\} \beta_0 = \beta, a_0 = a.$$  

(5)

Since calculating $Q(\beta, a)$ is NP-hard in general, there are different methods to estimate it [8], [13]. We propose a different approach.

As we described in Section I, we assume we have a (reasonably good) policy $\pi_o$ for the fully observable case and a corresponding $Q$ function $Q^{\pi_o}$. (Note that this is not the belief-MDP of the POMDP, but the MDP with the same dynamics and reward, just without the observation.)

This assumption is reasonable in a lot of applications, because solving the fully observable case is significantly easier. In a lot of cases, there are even procedural or analytic solutions. In absence of this, if the dynamics are known in a form that is easy to sample from, one can build a simulator and train a policy there [1].

Our other assumption is that the belief $\beta_t$ is multi-modal: it cannot be modelled as a mean value and some additive noise. This means simply guessing what the most probable state is at each timestep, and making a decision based on that is not enough.

Instead, we propose to define a POMDP policy $\pi_{po}$ based on the one-step estimation of the $Q$ function:

$$\tilde{Q}^{\pi_o}(\beta, a) := \mathbb{E}_{s \sim \beta} Q^{\pi_o}(s, a).$$  

(6)

Based on this definition, the POMDP policy can be defined as the greedy policy with respect to $\tilde{Q}^{\pi_o}$:

$$\pi_{po}(a \mid \beta) := \begin{cases} 1 & \text{if } a = \arg \max_{a' \in A} \tilde{Q}^{\pi_o}(\beta, a') \\ 0 & \text{otherwise} \end{cases}$$

(7)

(If we assume the maximum occurs for a single action or we choose one of the maximizing actions).

We call this estimation one-step, because, since $\pi_{po} \neq \pi_o$, $Q^{\pi_o}$ does not accurately represent the expected reward in the future, and we deal with the uncertainty coming from partial observability only in timestep $t$ when we take the expectation over $\beta_t$. In other words, $Q^{\pi_o}(\beta, a) \neq \tilde{Q}^{\pi_o}(\beta, a)$, even if they might be close for a certain problem class.

However, there is intuitive motivation behind this scheme, as it is the mathematical formulation of the question “What would a policy do that can see the state?”, weighted by our belief (probability) of each state.

Care must be taken when using a learned $Q^{\pi_o}$: usually, reinforcement learning algorithms learn from interactions between the agent and the environment, and the accuracy of information depends on which states (and actions) the agent visited during training. In particular, there might be states that the agent has never seen, and for which the $Q$ function is uninitialized.

This is normally not a problem if the same policy is used for training and inference, or when the $Q$ function is represented by powerful function approximators capable of generalization, but in (6), $Q^{\pi_o}$ must have valid values for every state with $\beta_t(s) > 0$. 

---

1To simplify discussion, we will assume that the discount factor $\gamma = 1$. The generalization to the discounted case is straightforward.
Another potential way to deal with this is to start the simulator from different states the agent can potentially reach.

III. EXPERIMENTS

A. Metrics used

To be able to analyze the fully-observable policy-based method, we used several metrics, the most straightforward of which is the average cumulated reward for each episode (the POMDP objective).

We used a "proxy policy" for every alternative policy that does not use the complete belief information: at each step, we chose the states with maximal belief:

\[ S_{\text{max}}(t) = \{ s \in S : \beta_t(s) = \max_{s'} \beta_t(s') \} , \]

taking into account that multiple states might be maximizing.

In this section, for notational convenience, we will implicitly assume that there may be multiple elements that maximize an \( \arg \max \) operator, so we define the \( \arg \max \) operator to be the set of maximizing elements:

\[ \arg \max_{x \in S} f(x) := \{ x \in S : f(x) = \max_{x'} f(x') \} . \]

Note that the assumption for this proxy policy is that it calculates the belief \( \beta_t \) for each timestep, but then discards information, retaining only the states with maximal probability. The intuition behind this is that the efficiency of this functions as an upper bound on any policy taking action based on any sort of state-identification method.

We compared how the actions taken by the alternative policy differ from our proposed policy: we computed the intersection-over-union (IoU) of the maximizing actions for each timestep. Formally, for each timestep, we define the action set from which the agent chooses one:

\[ A_{\text{mix}}(t) = \arg \max_{a \in A} \tilde{Q}(\beta_t, a) = \arg \max_{a \in A} \mathbb{E}_{s \sim \beta_t} Q(s, a) \]

("mix" denoting that this is defined using a mixture of Q values) and the probability-maximizing policy:

\[ A_{\text{max}}(t) = \arg \max_{a \in A} \mathbb{E}_{s \sim U} \tilde{Q}(s, a) , \]

where \( U \) denotes the uniform distribution. The IoU, or Jaccard index of these two action sets is:

\[ \text{IoU} = \frac{|A_{\text{max}} \cap A_{\text{mix}}|}{|A_{\text{max}} \cup A_{\text{mix}}|} . \]

Another metric we are interested in is the difference in value between the actions chosen by the two policies. Ideally, we would compare using the true value function \( Q(\beta, a) \), but since that is hard to compute, we again estimate it with the one step value function \( \tilde{Q}(\beta, a) \). That is, we measure the value margin against the alternative (benchmark) policy:

\[ M_{\text{alt}}(t) = \tilde{Q}(\beta_t, A_{\text{mix}}(t)) - \mathbb{E}_{a \sim U}(A_{\text{max}}(t)) \tilde{Q}(\beta_t, a) , \]

where we can write \( \tilde{Q}(\beta_t, A_{\text{mix}}(t)) \) since the value of \( \tilde{Q} \) is the same for all \( a \in A_{\text{mix}}(t) \) by definition. Since this metric is based on the estimation \( \tilde{Q}(\beta, a) \) and not on the true \( Q(\beta, a) \), comparing it to the difference in episode returns can be indicative of the estimation error.

B. Benchmark

For benchmark, we used a Reconnaissance Blind TicTacToe (RBT, inspired by Reconnaissance Blind Chess [9]–[11]).

In this task, the agent plays a game of TicTacToe, except it cannot see the moves of the opponent directly. Instead, before each move, a random subset of the cells is revealed. In this document, we will refer to this as sensing; we used rectangle-shaped sensing windows of different sizes.

Reward is 1 for winning, −1 for losing the game, −1 also for an invalid move, and 0 otherwise. In our implementation, invalid moves also terminate the episode.

On the one hand, this task has the defining features of a "hard" POMDP problem: the observation is not simply a noisy estimation of the state the agent: many possible (probable) but distinct states can be consistent with a given observation history, and to retain all information, has to remember observation potentially from the beginning of the episode.

On the other hand, the task is small enough that the belief can be calculated exactly using (3e); an example belief for \( t = 3 \) is shown in Fig. 1. This makes it possible to analyze the Q mixture-based policy without any corruption from an imperfect belief-estimator. A policy that solves the fully observable case is also easy to compute.

Note that the dynamics \( P(s_{t+1} \mid s_t, a_t) \) has two components: the agent action, which is deterministic (given \( a_t \)), and the opponent action, which is stochastic if the opponent policy is stochastic. Therefore, the uncertainty (in the belief) comes from both the incomplete observation and the stochastic dynamics, and by using different sense window sizes (controlling the amount of information from observation) and different opponent policies, we can control the amount and quality of the uncertainty in the environment.

There are two main differences from RBC: the opponent does not play blind, i.e., it observes the board state fully. This removes the belief of the opponent from the state [9], and frees the agent of any need to bluff. The second change is that in RBC, invalid actions (actions against the rules) resulted in no-operation actions, and since both players are playing blind, neither would have any unfair advantage because of this. In RBT, however, to simplify matters, we terminate the episode after an invalid action. This means that the belief-update can implicitly assume the action was valid, since otherwise the episode is terminated and the calculated belief is not used anyway.

C. Results

Fig. 2 and Table 2 show the average returns of our proposed policy, with the alternative policy that only takes the state with maximum probability into account (as described in Section III-A). As shown by the figure, our proposed policy consistently outperforms the alternative.
Fig. 1. Belief after 4 plies ($\beta_2$) in a randomly chosen episode with sensing window size $2 \times 2$. On the left are the observations with sensing windows marked with gray rectangles, on the right are the states with positive probability. First row contains the state visualizations, with their respective probabilities below. The true (hidden) state is the rightmost one.

Fig. 2. Average returns for the two policies across different sense window sizes. In this task, the reward is only given at the end of the episode, 1 for winning and $-1$ for losing, hence it is proportional to the win ratio. The measurements are based on 1000 episodes. Error bars indicate the 95% confidence estimates. Numerical values are shown in Table I.

| Sense window | Q-mixture policy (ours) | Alternative policy |
|--------------|-------------------------|--------------------|
| $1 \times 1$ | 0.215 ± 0.06            | -0.197 ± 0.06      |
| $2 \times 1$ | 0.316 ± 0.06            | 0.019 ± 0.06       |
| $2 \times 2$ | 0.532 ± 0.06            | 0.279 ± 0.06       |
| $3 \times 1$ | 0.592 ± 0.06            | 0.2 ± 0.06         |
| $3 \times 2$ | 0.813 ± 0.05            | 0.632 ± 0.05       |

To model the varying amount of information available in the observation, and with that, the number of probable states, we ran the experiment with different sensing window sizes. As expected, the larger the sensing window, the more information the model has, and the better the performance of both policies overall.

To gain more information about how well the policy works in this environment, we collected some metrics described in Section III-A, as shown in Fig. 3. As expected, at $t = 0$, there is only one possible state (the empty board), hence the two policies behave the same. At $t > 0$, however, the uncertainty in the state will cause the sets of actions taken by the two policies to diverge. Also, the best action (according to the expected Q values) is better by a significant margin than the others, particularly those of the alternative policy.

Since the average IoU remains below 1 in our experiments (as observable in Fig. 3), for a significant number of possible observation-action histories, an optimal action in a state of maximum probability is not included in the set of actions of the Q mixture-based policy. $M_{\text{alt}}$ is close to 0.2 throughout the episode in Fig. 3, which is comparable as the advantage in average return in Table I for the corresponding sense window size $2 \times 2$.

IV. CONCLUSION

We proposed a way to use policies solving the fully observable version of a POMDP to solve the POMDP itself. Our method is applicable for all problems where there is a solution for the fully observable version, and is especially useful when the belief is multimodal making decisions based on only one state is not enough.

We introduced a mathematical framework for our method and any further discussion on it, and designed an appropriate
benchmark to measure performance. We showed that our policy consistently outperforms the benchmark policy that only uses the state with maximum probability by 0.2−0.4 in average return.

For this work to be applicable in practice, the effects of imperfect belief estimation and also of the specific observation function on the performance of the agent have to be examined to determine the class of problems this technique is most applicable to. For example, when using a sampling-based belief reconstruction method, the number of samples needed for optimal decision should be analyzed.

We experimented in a small state space to be able to calculate the belief analytically. To scale our approach to more real and usable applications, a method of belief reconstruction will be needed.

REFERENCES

[1] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, 2nd ed. MIT press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[2] 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. [Online]. Available: https://doi.org/10.1038/nature14236

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” CoRR, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[4] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, and M. A. Riedmiller, “Maximum a posteriori policy optimisation,” in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=S1ANxQW0b

[5] E. J. Sondik, “The optimal control of partially observable markov processes over the infinite horizon: Discounted costs,” Operations Research, vol. 26, no. 2, pp. 282–304, 1978. [Online]. Available: http://www.jstor.org/stable/1609335

[6] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, “The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care,” Nature Medicine, vol. 24, no. 11, p. 1716, 2018.

[7] A. Mandel, Y. Liu, S. Levine, E. Brunskill, and Z. Popovic, “Offline policy evaluation across representations with applications to educational games,” in International conference on Autonomous Agents and Multi-Agent Systems, AAMAS ’14, Paris, France, May 5-9, 2014. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1077–1084. [Online]. Available: http://dl.acm.org/citation.cfm?id=2617417

[8] M. Hauskrecht, “Value-function approximations for partially observable markov decision processes,” J. Artif. Intell. Res., vol. 13, pp. 33–94, 2000. [Online]. Available: https://doi.org/10.1613/jair.678

[9] J. Markowitz, R. W. Gardner, and A. J. Llorens, “On the complexity of reconnaissance blind chess,” CoRR, vol. abs/1811.03119, 2018. [Online]. Available: http://arxiv.org/abs/1811.03119

[10] A. J. Newman, C. L. Richardson, S. M. Kain, P. G. Stankiewicz, P. R. Guesman, B. A. Schreurs, and J. A. Dunne, “Reconnaissance blind multi-chess: an experimentation platform for isr sensor fusion and resource management,” in Signal Processing, Sensor/Information Fusion, and Target Recognition XXV, vol. 9842. International Society for Optics and Photonics, 2016, p. 984209.

[11] R. W. Gardner, C. Lowman, C. Richardson, A. J. Llorens, J. Markowitz, N. Drenkow, A. Newman, G. Clark, G. Perrotta, R. Perrotta, T. Highley, V. Shcherbina, W. Bernadoni, M. Jordan, and A. Asenov, “The first international competition in machine reconnaissance blind chess,” in NeurIPS 2019 Competition and Demonstration Track, 8-14 December 2019, Vancouver, Canada. Revised selected papers, ser. Proceedings of Machine Learning Research, H. J. Escalante and R. Hudsell, Eds., vol. 123. PMLR, 2019, pp. 121–130. [Online]. Available: http://proceedings.mlr.press/v123/gardner20a.html

[12] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tvron, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, “Solving rubik’s cube with a robot hand,” CoRR, vol. abs/1910.07113, 2019. [Online]. Available: http://arxiv.org/abs/1910.07113

[13] H. Kurniawati, D. Hsu, and W. S. Lee, “SARSOFT: efficient point-based POMDP planning by approximating optimally reachable belief spaces,” in Robotics: Science and Systems IV, Eidgenössische Technische Hochschule Zürich, Zurich, Switzerland, June 25-28, 2008; O. Brock, J. Trinkle, and F. Ramos, Eds. The MIT Press, 2008. [Online]. Available: http://www.roboticsproceedings.org/rss04/p9.html

[14] T. S. Jaakkola, S. Singh, and M. I. Jordan, “Reinforcement learning algorithm for partially observable markov decision problems,” in Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994], G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds, MIT Press, 1994, pp. 345–352. [Online]. Available: http://papers.nips.cc/paper/951-reinforcement-learning-algorithm-for-partially-observable-markov-decision-problems

[15] D. Wierstra, A. Förster, J. Peters, and J. Schmidhuber, “Solving deep memory pomdps with recurrent policy gradients,” vol. 4668, 09 2007, pp. 697–706.

[16] M. Hauskrecht and P. Stone, “Deep recurrent q-learning for partially observable mdps,” in 2015 AAAI Fall Symposium Series, Jul. 2015.

[17] G. Clark, “Deep synaptic monyte-carlo planning in reconnaissance blind chess,” in Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 4106–4119. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/215a71e12769b656e3c3e27299f1c5e-d Abstract.html

[18] K. Blows and T. Highley, “An uncertainty exploration method in imperfect information games,” in Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=S1ANxQW0b