The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Learning Others' Intentional Models in Multi-Agent Settings Using Interactive POMDPs

Yanlin Han, Piotr Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Abstract

Interactive partially observable Markov decision processes (I-POMDPs) provide a principled framework for planning and acting in a partially observable, stochastic and multi-agent environment, extending POMDPs to multi-agent settings by including models of other agents in the state space and forming a hierarchical belief structure. In order to predict other agents' actions using I-POMDP, we propose an approach that effectively uses Bayesian inference and sequential Monte Carlo (SMC) sampling to learn others' intentional models which ascribe to them beliefs, preferences and rationality in action selection. Empirical results show that our algorithm accurately learns models of other agents and has superior performance when compared to other methods. Our approach serves as a generalized reinforcement learning algorithm that learns other agents' beliefs, and transition, observation and reward functions. It also effectively mitigates the belief space complexity due to the nested belief hierarchy.

Introduction

Partially observable Markov decision processes (POMDPs) (Kaelbling, Littman, and Cassandra 1998) provide a principled, decision-theoretic framework for planning under uncertainty in a partially observable, stochastic environment. An autonomous agent operates rationally in such settings by maintaining a belief of the physical state at any given time, in doing so it sequentially chooses the optimal actions that maximize the expected value of future rewards. Although POMDPs can be used in multi-agent settings, doing so treats other agents' actions as noise and folds the effects of their actions into the state transition function, such as recent Bayes-adaptive POMDPs (Ross, Chaib-draa, and Pineau 2008), infinite generalized policy representation (Liu, Liao, and Carin 2011), and infinite POMDPs (Doshi-Velez et al. 2015). Therefore, an agent's beliefs about other agents are not part of the solutions of POMDPs.

Interactive POMDPs (I-POMDPs) (Gmytrasiewicz and Doshi 2005) are a generalization of POMDP to multi-agent settings. They replace POMDP belief spaces with interactive hierarchical belief systems. Specifically, an I-POMDP augments the plain beliefs about the physical states in POMDP by including models of other agents, which forms a hierarchical belief structure that represents an agent's belief about the physical state, belief about the other agents and their beliefs about others' beliefs, and so on. The models of other agents included in the new augmented state space consist of two types: intentional models and subintentional models. An intentional model ascribes beliefs, preferences, and rationality to other agents (Gmytrasiewicz and Doshi 2005), while a simpler subintentional model, such as finite state controllers (Panella and Gmytrasiewicz 2017) (Panella and Gmytrasiewicz 2016), does not. Solutions of I-POMDPs map an agent's belief about the environment and other agents' models to actions. It has been shown (Gmytrasiewicz and Doshi 2005) that the added sophistication of modeling others as rational agents results in a higher value function compared to one obtained from treating others as noise, which implies the modeling superiority of I-POMDPs for multi-agent systems over other approaches.

However, the interactive belief modification for I-POMDPs results in a drastic increase of the belief space complexity, adding to the curse of dimensionality: the complexity of the belief representation is proportional to belief dimensions due to exponential growth of agent models with increase of nesting level. Since exact solutions to POMDPs are proven to be PSPACE-complete for finite time horizon and undecidable for infinite time horizon (Papadimitriou and Tsitsiklis 1987), the time complexity of more generalized I-POMDPs, which may contain multiple POMDPs or I-POMDPs of other agents, is at least PSPACE-complete for finite horizon and undecidable for infinite horizon. Due to this complexity, a solution which accounts for an agent's belief over an entire intentional model has not been implemented up to date. There are partial solutions that depend on what is known about other agents' beliefs about the physical states (Doshi and Gmytrasiewicz 2009), but they do not include the state of an agent's knowledge about others' reward, transition, and observation functions. Indirect approach such as subintentional finite state controllers (Panella and Gmytrasiewicz 2017) (Panella and Gmytrasiewicz 2016) does not include any of these elements either. To unleash the full modeling power of intentional models and to apply I-POMDPs to realistic settings, a robust approximation algorithm for computing the nested interactive belief and predicting other agents' actions is crucial to the trade-off between solution quality and computation complexity.

To address this issue, we propose a Bayesian approach that utilizes customized sequential Monte Carlo sampling (De Freitas, Doucet, and Gordon 2001) to obtain approximate solutions to I-POMDPs and implement the algorithms in a software package.[1] We assume that agents maintain beliefs over intentional models of other agents and make sequential Bayesian updates using observations from the environment. Since this Bayesian inference task is analytically intractable due to the need of computing high dimensional integration, we devise a customized sequential Monte Carlo method to descend the belief hierarchy, parametrize others' models and sample all model parameters at each nesting level, starting from the interactive particle filter (I-PF) (Doshi and Gmytrasiewicz 2009) for I-POMDP belief update.

Recently there has been research progress on modeling and learning other agents' models in multi-agent systems, but none of them have managed to learn over the entire space of others' models in the formulation of multi-agent POMDPs. In particular, a previous work of Bayes Adaptive I-POMDPs (BA-IPOMDPs) (Ng et al. 2012) incorporate model learning in I-POMDPs by modeling transition and observation functions using additional Dirichlet distributions. However, the BA-IPOMDP does not learn the reward function, which is a key component of I-POMDP. In some opponent modeling approaches, Bayesian Policy Reuse have been used to learn opponents' policies in MDP settings. For instance, in (Hernandez-Leal et al. 2016) and (Hernandez-Leal and Kaisers 2017), they both have mixed online and offline learning methods in MDP settings that combine game and decision theoretic approaches to directly learn others' policies, while our method learns others' models and computes their strategies as needed, which focuses on a concise, decision theoretic framework in POMDP settings.

Our approach, for the first time, successfully recovers others' models over the entire intentional model space which contains their beliefs, and transition, observation and reward functions, making it a generalized reinforcement learning method for multi-agent settings. By approximating Bayesian inference using a customized sequential Monte Carlo sampling method, we significantly mitigate the belief space complexity of I-POMDPs.

Background

POMDP

A Partially observable Markov decision process (POMDP) (Kaelbling, Littman, and Cassandra 1998) is a general reinforcement learning model for planning and acting in a single-agent, partially observable, stochastic domain. It is defined for a single agent i as:

$$POMDP_i = \langle S, A_i, \Omega_i, T_i, O_i, R_i \rangle \tag{1}$$

Where the meaning for each element in the 6-tuple is:

- S is the set of states of the environment.
- A_i is the set of agent i's possible actions
- Ω_i is the set of agent i's possible observations
- $T_i : S \times A_i \times S \rightarrow [0, 1]$ is the state transition function
- $O_i : S \times A_i \times \Omega_i \rightarrow [0, 1]$ is the observation function
- $R_i : S \times A_i \rightarrow \mathbb{R}$ is the reward function.

Given the definition above, an agent's belief about the state can be represented as a probability distribution over S. The belief update can be simply done using the following formula, where α is the normalizing constant:

$$b(s') = \alpha O(o, s, a) \sum_{s \in S} T(s', a, s) b(s) \tag{2}$$

Conveniently, the equation above can be summarized as $b(s') = SE(b, a, o)$.

To quantify the value of a belief state, we can associate the utility with a belief state b_i, which is composed of the best immediate reward and the discounted expected sum of utilities of the following belief states:

$$\begin{aligned} U(b_i) =& \max_{a_i \in A_i} \Big\{ \sum_{s \in S} b_i(s) R(s, a_i) \\ &+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_i) \times U(SE(b_i, a_i, o_i)) \Big\} \end{aligned} \tag{3}$$

Then the optimal action, a^*, is simply part of the set of optimal actions, $OPT(b_i)$, for the belief state defined as:

$$\begin{aligned} OPT(b_i) =& \arg\max_{a_i \in A_i} \Big\{ \sum_{s \in S} b_i(s) R(s, a_i) \\ &+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_i) \times U(SE(b_i, a_i, o_i)) \Big\} \end{aligned} \tag{4}$$

Particle Filter

The Markov Chain Monte Carlo (MCMC) method (Gilks, Richardson, and Spiegelhalter 1996) is widely used to approximate probability distributions that are difficult to compute directly. MCMC generates samples from a posterior distribution $\pi(x)$ over state space x, by simulating a Markov chain $p(x'|x)$ whose state space is x and stationary distribution is $\pi(x)$. The samples drawn from p converge to the target distribution π as the number of samples goes to infinity.

In order to make MCMC work on sequential inference task, especially sequential decision making under Markov assumption, sequential Monte Carlo (SMC) methods have been proposed and some of them are capable of dealing with high dimensionality and/or complexity problems, such as particle filters (Del Moral 1996). At each time step, a particle filter draws samples (or particles) from a proposal distribution, commonly the conditional distribution $p(x_t|x_{t-1})$ of the current state x_t given the previous x_{t-1}, then use the observation function $p(y_t|x_t)$ to compute importance weights for all particles and resample them according to the weights.

[1] https://github.com/solohan22/I-POMDP

The Model

I-POMDP framework

An interactive POMDP of agent i (Gmytrasiewicz and Doshi 2005), I-POMDP i, is defined as:

$$I\text{-}POMDP_{i,l} = \langle IS_{i,l}, A, \Omega_i, T_i, O_i, R_i \rangle \tag{5}$$

where $IS_{i,l}$ is a set of interactive states, defined as $IS_{i,l} = S \times M_{j,l-1}, l \geq 1$, where S is the set of physical states and $M_{j,l-1}$ is the set of possible models of agent j, and l is the strategy (nesting) level.

A specific class of models are the $(l-1)$th level *intentional* models, $\Theta_{j,l-1}$, of agent j: $\theta_{j,l-1} = \langle b_{j,l-1}, A, \Omega_j, T_j, O_j, R_j, OC_j \rangle$, where $b_{j,l-1}$ is agent j's belief nested to the level $(l-1)$, $b_{j,l-1} \in \Delta(IS_{j,l-1})$, and OC_j is j's optimality criterion. The intentional model $\theta_{j,l-1}$, analogous to *type* as used in Bayesian games (Harsanyi 1967), can be rewritten as $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta_j} \rangle$, where $\hat{\theta_j}$ includes all elements of the intentional model other than the belief and is called the agent j's frame.

As discussed in (Gmytrasiewicz and Doshi 2005), the *subintentional* models constitute the remaining models in $M_{j,l-1}$, examples of subintentional models are finite state controllers (Panella and Gmytrasiewicz 2016) and fictitious play models (Fudenberg and Levine 1998). In this paper, we focus on intentional models and do not try to fully address subintentional models.

The $IS_{i,l}$ could be defined in an inductive manner:

$$\begin{aligned} &IS_{i,0} = S, && \theta_{j,0} = \{\langle b_{j,0}, \hat{\theta_j} \rangle : b_{j,0} \in \Delta(S)\} \\ &IS_{i,1} = S \times \theta_{j,0}, && \theta_{j,1} = \{\langle b_{j,1}, \hat{\theta_j} \rangle : b_{j,1} \in \Delta(IS_{j,1})\} \\ &\ldots\ldots && \\ &IS_{i,l} = S \times \theta_{j,l-1}, && \theta_{j,l} = \{\langle b_{j,l}, \hat{\theta_j} \rangle : b_{j,l} \in \Delta(IS_{j,l})\} \end{aligned} \tag{6}$$

All remaining components in an I-POMDP are similar to those in a POMDP, the major difference is that they also involve other agents' actions:

- $A = A_i \times A_j$ is the set of joint actions of all agents.
- Ω_i is the set of agent i's possible observations.
- $T_i : S \times A \times S \to [0,1]$ is the state transition function.
- $O_i : S \times A \times \Omega_i \to [0,1]$ is the observation function.
- $R_i : IS \times A \to \mathbb{R}$ is the reward function.

Interactive belief update

Given all the definitions above, the interactive belief update can be performed as follows, by considering others' actions and anticipated observations:

$$\begin{aligned} &b_i^t(is^t) = Pr(is^t|b_i^{t-1}, a_i^{t-1}, o_i^t) \\ &= \alpha \sum_{is^{t-1}} b(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) T(s^{t-1}, a^{t-1}, s^t) \\ &\times O_i(s^t, a^{t-1}, o_i^t) \sum_{o_j^t} O_j(s^t, a^{t-1}, o_j^t) \tau(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) \end{aligned} \tag{7}$$

Compared with POMDP, the interactive belief update in I-POMDP takes two additional sophistications into account. First, the probability of other's actions given his models needs to be computed since the state now depends on both agents' actions (the second summation). Second, the modeling agent needs to update his beliefs based on the anticipation of what observations the other agent might get and how it updates (the third summation).

Similarly to POMDPs, the utilities associated with a belief state in I-POMDPs can be updated as:

$$\begin{aligned} U(\theta_i) = \max_{a_i \in A_i} \Big\{ &\sum_{is \in IS} b_{is}(s) ER_i(is, a_i) \\ &+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \hat{\theta_i} \rangle) \Big\} \end{aligned} \tag{8}$$

where $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j|\theta_j)$.

Then the optimal action, a_i^*, for an infinite horizon criterion with discounting, is part of the set of optimal actions, $OPT(\theta_i)$, for the belief state, defined as:

$$\begin{aligned} OPT(\theta_i) = \arg\max_{a_i \in A_i} \Big\{ &\sum_{is \in IS} b_{is}(s) ER_i(is, a_i) \\ &+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \hat{\theta_i} \rangle) \Big\} \end{aligned} \tag{9}$$

Sampling Algorithms

Description

The Interactive Particle Filter (I-PF) (Doshi and Gmytrasiewicz 2009) was proposed as a filtering algorithm for interactive belief update in I-POMDP. It generalizes the classic particle filter algorithm to multi-agent settings and uses the state transition function as the proposal distribution, which is usually used in a specific particle filter algorithm called bootstrap filter (Gordon, Salmond, and Smith 1993). However, due to the enormous belief space, the I-PF implementation assumes that the other agent's frame $\hat{\theta_j}$ is known to the modeling agent, simplifying the belief update from $S \times \Theta_{j,l-1}$ to a significantly smaller space $S \times b_{j,l-1}$.

Our interactive belief update described in Algorithm 1, however, generalizes I-PF to the entire intentional model space, and this generalization is nontrivial. First, in order to update the belief over intentional model space of other agents, the set of N initial belief samples $\theta_{-k}^{(n),t-1} =< b_{-k}^{(n),t-1}, A_{-k}, \Omega_{-k}, T_{-k}^{(n)}, O_{-k}^{(n)}, R_{-k}^{(n)}, OC_{-k} >$, where k here denotes the modeling agent and $-k$ denotes all other modeled agents. We assume that the actions A_{-k}, observations Ω_{-k} and optimality criteria OC_k are known to all agents. Second, the observation function of the modeled agent(s), $O_{-k}^{(n)}(o_{-k}^t|s^{(n),t}, a_k^{t-1}, a_{-k}^{t-1})$, in line 13 is now randomized as a consequent, since it is not assumed to be known by the modeling agent. Lastly, we add another resampling step in line 18 in order to avoid divergence due to dramatic increase of the sampling space, by resampling each

Algorithm 1: Interactive Belief Update

```
$\tilde{b}^t_{k,l}$ = InteractiveBeliefUpdate($\tilde{b}^{t-1}_{k,l}$, $a^{t-1}_k$, $o^t_k$, $l > 0$)
1  For $is^{(n),t-1}_k =< s^{(n),t-1}, \theta^{(n),t-1}_{-k} >\in \tilde{b}^{t-1}_{k,l}$,
2      sample $a^{t-1}_{-k} \sim P(A_{-k}|\theta^{(n),t-1}_{-k})$
3      sample $s^{(n),t} \sim T_k(S^t|S^{(n),t-1}, a^{t-1}_k, a^{t-1}_{-k})$
4      for $o^t_{-k} \in \Omega_{-k}$:
5          if $l = 1$:
6              $b^{(n),t}_{-k,0}$ = Level0BeliefUpdate($b^{(n),t-1}_{-k,0}$, $a^{t-1}_{-k}$,
                       $o^t_{-k}, \theta^{(n),t-1}_{-k}$)
7              $\theta^{(n),t}_{-k} =< b^{(n),t}_{-k,0}, \hat{\theta}^{(n),t-1}_{-k} >$
8              $is^{(n),t}_k =< s^{(n),t}, \theta^{(n),t}_{-k} >$
9          else:
10             $b^{(n),t}_{-k,l-1}$ = InteractiveBeliefUpdate($\tilde{b}^{t-1}_{-k,l-1}$,
                       $a^{t-1}_{-k}, o^t_{-k}, l-1$)
11             $\theta^{(n),t}_{-k} =< b^{(n),t}_{-k,l-1}, \hat{\theta}^{(n),t-1}_{-k} >$
12             $is^{(n),t}_k =< s^{(n),t}, \theta^{(n),t}_{-k} >$
13         $w^{(n)}_t = O^{(n)}_{-k}(o^t_{-k}|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
14         $w^{(n)}_t = w^{(n)}_t \times O_k(o^t_k|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
15         $\tilde{b}^{temp}_{k,l} =< is^{(n),t}_k, w^{(n)}_t >$
16 normalize all $w^{(n)}_t$ so that $\sum_{n=1}^N w^{(n)}_t = 1$
17 resample from $\tilde{b}^{temp}_{k,l}$ accroding to normalized $w^{(n)}_t$
18 resample $\theta^{(n),t}_{-k}$ according to similar neighboring models
19 return $\tilde{b}^t_{k,l} = is^{(n),t}_k$
```

dimension of the model samples from a Gaussian distribution with the mean of current sample value.

The Algorithm 1 starts from a set of initial prior samples $is^{(n),t-1}_k$. For each of $is^{(n),t-1}_k$, it samples other agents' optimal actions a^{t-1}_{-k} from $P(A_{-k}|\theta^{(n),t-1}_{-k})$ obtained from POMDP solver Perseus[2] (Spaan and Vlassis 2005). Then it samples the physical state s^t using the state transition $T_k(S^t|S^{(n),t-1}, a^{t-1}_k, a^{t-1}_{-k})$. Once a^{t-1}_{-k} and s^t are sampled, the algorithm calls the 0-level belief update (line 5 to 8), described in Algorithm 2, to update other agents' beliefs over physical state $b^t_{-k,0}$ if the current nesting level l is 1; or recursively calls itself at a lower level $l-1$ (line 9 to 12) if the current nesting level is greater than 1. The sample weights $w^{(n)}_t$ are computed according to observation likelihoods of both modeling and modeled agents (line 13, 14), and then normalized (line 16). Lastly, the algorithm resamples the intermediate samples according to the computed weights (line 17) and resamples another time from similar neighboring models (line 18) to avoid divergence.

Algorithm 2: Level-0 Belief Update

```
$b^t_{k,0}$ =Level0BeliefUpdate($b^{t-1}_{k,0}$,$a^{t-1}_k$,$o^t_k$, $T^{(n)}_k$,$O^{(n)}_k$)
1  $P(a^{t-1}_{-k}) = 1/a^{t-1}_{-k}$
2  for $s^t \in S$:
3      for $s^{t-1}$:
4          for $a^{(t-1)}_{-k} \in A_{-k}$:
5              $P^{(n)}(s^t|s^{t-1}, a^{t-1}_k) =$
                       $T^{(n)}_k(s^t|s^{t-1}, a^{t-1}_k, a^{t-1}_{-k})P(a^{t-1}_{-k})$
6          $sum^{(n)} += P^{(n)}(s^t|s^{t-1}, a^{t-1}_k)b^{t-1}_{k,0}(s^{t-1})$
7      for $a^{(t-1)}_{-k} \in A_{-k}$:
8          $P^{(n)}(o^t_k|s^t, a^{t-1}_k) +=$
                       $O^{(n)}_k(o^t_k|s^t, a^{t-1}_k, a^{t-1}_{-k})P(a^{t-1}_{-k})$
9      $b^t_{k,0} = sum^{(n)}P^{(n)}(o^t_k|s^t, a^{t-1}_k)$
10 normalize and return $b^t_{k,0}$
```

Consequently, the 0-level belief update, described in Algorithm 2, treats other agents' actions as noise, randomizes the state transition and observation functions, and input them as arguments. For each possible action a^{t-1}_{-k}, it computes the actual state transition (line 5) and actual observation function (line 8) by marginalizing over others' actions, and returns the normalized belief $b^t_{k,0}$. Notice that the transition function $T^{(n)}_k(s^t|s^{t-1}, a^{t-1}_k, a^{t-1}_{-k})$ and observation function $O^{(n)}_k(o^t_k|s^t, a^{t-1}_k, a^{t-1}_{-k})$ are both samples from input arguments, which depend on particular model parameters of the actual agent on the 0th level.

Illustration

We illustrate the operations of Algorithm 1 and 2 using the multi-agent version of tiger problem (Gmytrasiewicz and Doshi 2005). The multi-agent tiger game is a generalization of the classical single agent tiger game (Kaelbling, Littman, and Cassandra 1998). It contains additional observations caused by others' actions, and the transition and reward functions involve others' actions as well.

For the illustrative simplicity, we assume there are two agents i and j, and the nesting level is 1. Recall that an interactive POMDP of agent i is defined as a six tuple $I\text{-}POMDP_i = \langle IS_{i,l}, A, \Omega_i, T_i, O_i, R_i \rangle$, thus for the specific setting of two-agent tiger problem:

- $IS_{i,1} = S \times \theta_{j,0}$, where $S = \{$tiger on the left (TL), tiger on the right (TR)$\}$ and $\theta_{j,0} =< b_j(s), A_j, \Omega_j, T_j, O_j, R_j, OC_j >\}$.
- $A = A_i \times A_j$ is a combination of both agents' possible actions: listen (L), open left door (OL) and open right door(OR).
- Ω_i: $\{$growl from left (GL) or right (GR)$\} \times \{$creak from left (CL), right (CR) or silence (S)$\}$.
- $T_i = T_j : S \times A_i \times A_j \times S \to [0, 1]$.
- $O_i : S \times A_i \times A_j \times \Omega_i \to [0, 1]$.
- $R_i : IS \times A_i \times A_j \to \mathbb{R}$.

Figure 1 illustrates the interactive belief update in the game described above. Suppose the sample size is 8, each dot represents a particular belief sample and the subscripts denotes the corresponding agents. The propagation step is

[2] http://www.st.ewi.tudelft.nl/~mtjspaan/pomdp/index_en.html

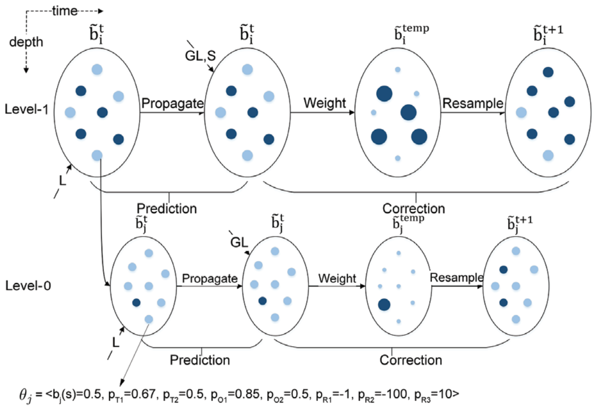

Figure 1: An illustration of interactive belief update for two agents and level-1 nesting.

implemented in lines 2 to 12 in Algorithm 1, the weighting step is in lines 13 to 16, and the resampling step is in lines 17 and 18. The belief update for a particular level-0 model sample, $\theta_j = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10\rangle$, is solved using Algorithm 2, and the optimal action is computed by calling the Perseus POMDP solver.

Experiments

To demonstrate the soundness of our theoretical framework, we present the results using the multi-agent tiger game (Gmytrasiewicz and Doshi 2005) with various settings. For the sake of brevity we restrict the experiments to a two-agent setting and level-1 nesting, but the sampling algorithm is extensible to any number of agents and nesting levels in a straightforward manner according to Algorithm 1.

Parameter Space

For the experiments of two-agent tiger game, we want to learn over all possible intentional models of the other agent j: $\theta_j =< b_j(s), A_j, \Omega_j, T_j, O_j, R_j, OC_j >$. As mentioned before we assume that A_j and Ω_j are known, and OC_j is infinite horizon with discounting. We want to recover the possible initial belief b_j^0 about the physical state, the transition, T_j, the observation, O_j and the reward, R_j, which can be all parametrized by eight parameters as shown in Table 1. We see that it is a large 8-dimensional parameter space to learn from: $b_j^0 \times p_{T1} \times p_{T2} \times p_{O1} \times p_{O2} \times p_{R1} \times p_{R2} \times p_{R3}$, where $\{b_j, p_T 1, p_T 2, p_O 1, p_O 2\} \in [0, 1] \subset \mathbb{R}$ and $\{p_R 1, p_R 2, p_R 3\} \in [-\infty, +\infty]$.

Results

We fix the number of samples to 2000 and run experiments with agent j acting according to three different policies shown in Figure 2. For brevity we focus on results of learning models when agent j acts according to the first of these policies, but give a performance comparison among all of them.

The aim of first experiment is trying to learn models of agent j who is modeling his opponent using a subintentional model. Agent j's actual policy, as shown in Figure 2(a), is to look for three consecutive growls from the same direction and then open the opposing door. The second experiment involves agent j equipped with high listening accuracy of 0.95 and small penalty of -10 for encountering the tiger, i.e. the agent j alternately opens door and listens as shown in Figure 2(b). And the third experiment involves a simple agent j who always listens since the listening penalty is now equal to the reward, as shown in Figure 2(c). In conclusion, one can view the difficulties of learning such agents' models as relatively hard, medium, and easy, since the policy difficulties decrease in experiment one, two and three. Meanwhile the parameters being learned will be less definite from experiment one to three, since there are more possible models which can generate the same policy when it becomes easier.

Table 1: Parameters for transition, observation and reward functions of two-agent tiger game

S	A	TL	TR
TL	L	p_{T1}	$1 - p_{T1}$
TR	L	$1 - p_{T1}$	p_{T1}
*	OL	p_{T2}	$1 - p_{T2}$
*	OR	$1 - p_{T2}$	p_{T2}

S	A	GL	GR
TL	L	p_{O1}	$1 - p_{O1}$
TR	L	$1 - p_{O1}$	p_{O1}
*	OL	p_{O2}	$1 - p_{O2}$
*	OR	$1 - p_{O2}$	p_{O2}

S	A	R
*	L	p_{R1}
TL	OL	p_{R2}
TR	OR	p_{R2}
TL	OR	p_{R3}
TR	OL	p_{R3}

To demonstrate the learning ability of our algorithm for possible models of the agent in Figure 2(a), we assign uninformative prior distributions to each parameter space , which is shown in Figure 3. They are uniform distributions: $\{b_j^0, p_{T1}, p_{T2}, p_{O1}, p_{O2}\} \sim U(0, 1)$, $\{p_{R1}, p_{R2}, p_{R3}\} \sim U(-200, 200)$. After 50 time steps, the algorithm converges to a posterior distribution over agent j's intentional models. From the marginal distributions of all parameters, we can see that the majority of samples are centered around the true parameter values.

Since the original parameter space is 8-dimensional, in order to visualize the learning process, we use principal component analysis (PCA) (Abdi and Williams 2010) to reduce it to 2D and plot it out as a 3D histogram, as shown in Figure 4. It starts from an uninformative prior and gradually converges to the most likely models. Eventually the mean value of this cluster ⟨0.49, 0.69, 0.49, 0.82, 0.51, -0.95, -99.23, 10.09⟩ is very close to the actual model ⟨0.5, 1, 0.5, 0.95, 0.5, -1, -10, 10⟩.

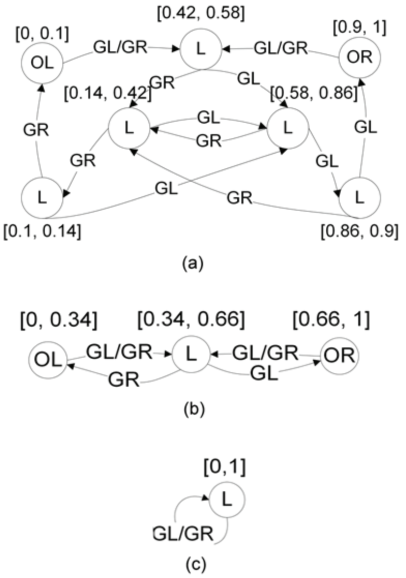

Figure 2: Optimal policies denoted as Finite State Controllers (FSCs) of: (a) $\theta_{j1} =< 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10 >$, (b) $\theta_{j2} = \langle$ 0.5, 1, 0.5, 0.95, 0.5, -1, -10, 10 $\rangle$, and (c) $\theta_{j3} = \langle$0.5, 0.66, 0.5, 0.85, 0.5, 10, -100, 10$\rangle$.

In Figure 5 we show that the learning quality in terms of KL-Divergence, which measures the distance between the learned mean values of model parameters and corresponding ground truth, becomes better as the number of particles increases in all three experiments.

Because agent i is now able to learn others' likely models, he should be capable of predicting j's actions relatively accurately. Therefore, we tested the performance of our algorithm in terms of prediction accuracy towards others' actions, which is the number of incorrect predictions with respect to others' actions over the ground truth. For conciseness, we show the average prediction error rates for all three experiments in Figure 6. We compared the results with other modeling approaches, such as a frequency-based (fictitious play) (Fudenberg and Levine 1998) approach, in which agent j is assumed to choose his action according to a fixed but unknown distribution, and a no-information model, in which agents assume others' actions are drawn from a uniform distribution and therefore is an instance of subintentional model. The shown results are averaged plots of 10 random runs, each of which has 50, 30 and 30 time steps respectively. It shows that the intentional I-POMDP approach has significantly lower error rates as agent i perceives more observations. The subintenional model assumes j's action is draw from a uniform distribution, therefore has a fixed high error rate. The frequency based approach has certain learning ability but is less sophisticated for modeling a rational agent, therefore its performance falls somewhere in between.

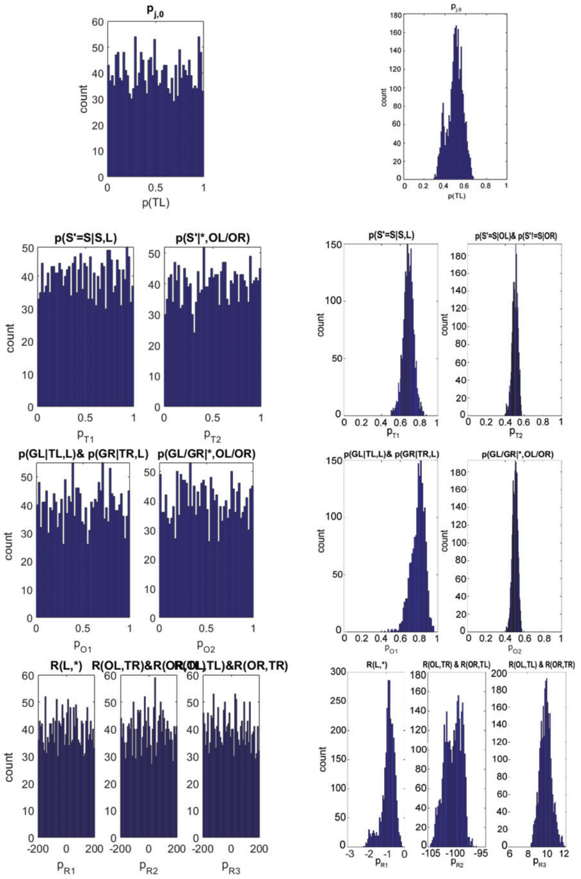

Figure 3: Assigned uniform priors (left) and learned posterior distributions (right) over model parameters for $\theta_{j1} = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10\rangle$. The means of learned posteriors are 0.49, 0.69, 0.49, 0.82, 0.51, -0.95, -99.23, 10.09.

Conclusions and Future Work

We have described a new approach to learn other agents' intentional models by approximating the interactive belief update using Bayesian inference and Monte Carlo sampling methods. We show the soundness of our theoretical framework using a multi-agent tiger game in which it correctly learns others' models over the entire intentional model space and can be generalized to problems of larger scale in a straightforward manner. Therefore, it provides a generalized reinforcement learning algorithm for multi-agent settings.

For future research opportunities, the applications presented in this paper can be extended to more complicated multi-agent problems. Due to computational complexity, experiments on higher nesting levels are currently limited, more efforts can be made on leveraging nonparametric Bayesian methods which inherently deal with nested belief structures. Besides, deep reinforcement learning methods which utilize various deep neural networks to approximate key components in POMDPs should also be capable of approximating corresponding functions in I-POMDPs, thus

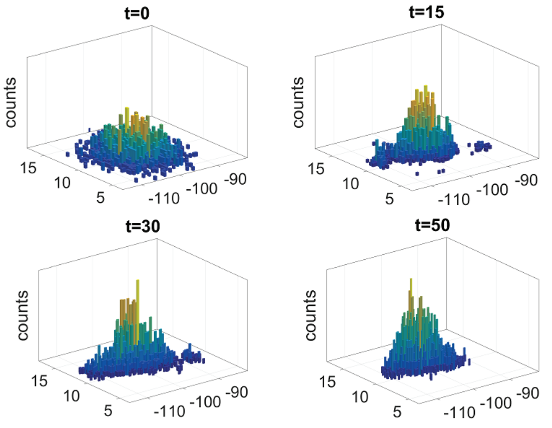

Figure 4: Histogram of all model samples for θ_{j1} during learning, after projection from 8D to 2D.

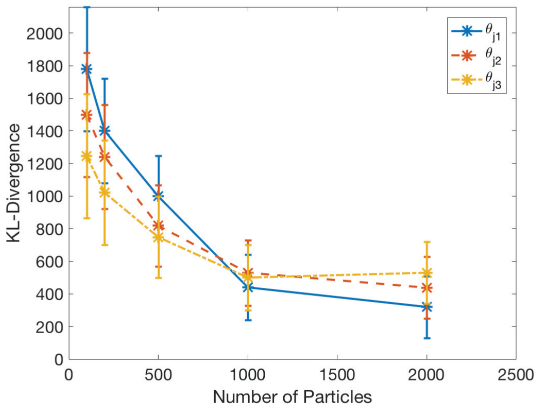

Figure 5: Learning quality measured by KL-Divergence,. The vertical bars are the standard deviations.

has potential of making the computations more efficient.

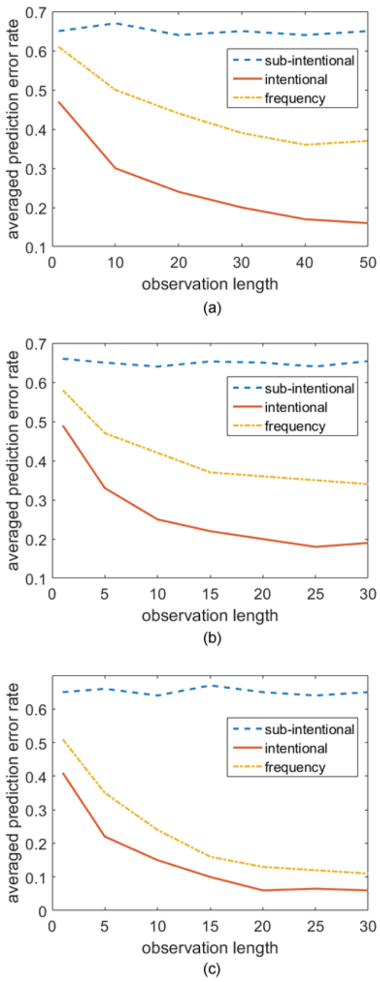

Figure 6: Performance comparisons in terms of prediction error rate vs observation length for (a) $\theta_{j1} = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10\rangle$ (b) $\theta_{j2} = \langle$ 0.5, 1, 0.5, 0.95, 0.5, -1, -10, 10 $\rangle$, and (c) $\theta_{j3} = \langle$0.5, 0.66, 0.5, 0.85, 0.5, 10, -100, 10$\rangle$.

References

Abdi, H., and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2(4):433–459.

De Freitas, N.; Doucet, A.; and Gordon, N. 2001. An introduction to sequential monte carlo methods. *SMC Practice. Springer Verlag*.

Del Moral, P. 1996. Non-linear filtering: interacting particle resolution. *Markov processes and related fields* 2(4):555–581.

Doshi, P., and Gmytrasiewicz, P. J. 2009. Monte carlo sampling methods for approximating interactive pomdps. *Journal of Artificial Intelligence Research* 34:297–337.

Doshi-Velez, F.; Pfau, D.; Wood, F.; and Roy, N. 2015. Bayesian nonparametric methods for partially-observable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence* 37(2):394–407.

Fudenberg, D., and Levine, D. K. 1998. *The theory of learning in games*, volume 2. MIT press.

Gilks, W. R.; Richardson, S.; and Spiegelhalter, D. J. 1996. Introducing markov chain monte carlo. *Markov chain Monte Carlo in practice* 1:19.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *J. Artif. Intell. Res.(JAIR)* 24:49–79.

Gordon, N. J.; Salmond, D. J.; and Smith, A. F. 1993. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, 107–113. IET.

Harsanyi, J. C. 1967. Games with incomplete information played by bayesian players, i–iii: part i. the basic model&. *Management science* 14(3):159–182.

Hernandez-Leal, P., and Kaisers, M. 2017. Towards a fast detection of opponents in repeated stochastic games. In *1st Workshop on Transfer in Reinforcement Learning at AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems.

Hernandez-Leal, P.; Rosman, B.; Taylor, M. E.; Sucar, L. E.; and Munoz de Cote, E. 2016. A bayesian approach for learning and tracking switching, non-stationary opponents. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 1315–1316. International Foundation for Autonomous Agents and Multiagent Systems.

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

Liu, M.; Liao, X.; and Carin, L. 2011. The infinite regionalized policy representation. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 769–776.

Ng, B.; Boakye, K.; Meyers, C.; and Wang, A. 2012. Bayes-adaptive interactive pomdps. In *AAAI*.

Panella, A., and Gmytrasiewicz, P. J. 2016. Bayesian learning of other agents' finite controllers for interactive pomdps. In *AAAI*, 2530–2536.

Panella, A., and Gmytrasiewicz, P. 2017. Interactive pomdps with finite-state models of other agents. *Autonomous Agents and Multi-Agent Systems* 1–44.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of markov decision processes. *Mathematics of operations research* 12(3):441–450.

Ross, S.; Chaib-draa, B.; and Pineau, J. 2008. Bayes-adaptive pomdps. In *Advances in neural information processing systems*, 1225–1232.

Spaan, M. T., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research* 24:195–220.