Pursuit-Evasion Game with an Agent Unaware of its Role

Jeff Park

***Abstract*—This paper uses a pursuit-evasion game involving a single agent and a single target to study how an agent unaware of its role (i.e. pursuer or evader) can still make an optimal decisions by inferring the role of the target. The game is modeled as a partially observable Markov decision process (POMDP), and the state belief was updated using a Bayesian parameter learning in which the parameter to infer is the probability that the agent is a pursuer. The model was solved using QMDP, and simulations were run to gauge the performance of the POMDP model and the suggested belief updater. It was shown that the belief updater works moderately well for estimating the agent's role. It was also shown that in POMDP model the agent performs extra exploratory steps to update its belief of its role. In simulations, the agent successfully captured and evaded the target in both pursuit and evasion games.**

I. Introduction

When making any decision, an agent must gather necessary data to be able to make an informed decision. However, the data, whether it is from sensors or dynamics model of the system, is always prone to uncertainties in the real world. For example, a robot trying to locate its position with sensors can never acquire its exact position due to noises, or uncertainties, inherent to its sensors. To mitigate the uncertainties, state estimation techniques such as particle filter and Kalman filter have been developed and studied to provide approximate information on the robot's whereabout based on noisy observations [1].

On other occasions, an agent must make decisions while interacting with other agents, such as humans. For example, Sundberg et al. studied autonomous lane changing on the freeway scenario in which an agent tried to infer the internal state (i.e. aggressiveness) of other human drivers [2]. The internal state was modeled as a partially observable Markov decision process (POMDP), and the result showed that when internal states are correlated, it presented a better performance than the baseline. In another work, Egorov et al. studied target surveillance under the presence of ballistic threat [3]. In this work, the adversary was modeled using level-k policies, a behavioral model that involves recursive reasoning over k levels. The problem was modeled as a mixed observability Markov decision process (MOMDP), and the result showed that level-3 policies outperformed lower-level policies and human players. Hoang and Low also showed that an MDP involving an agent of level-k reasoning can be solved using nested MDP [4] to predict the intention and strategies of the the agent.

In this project, a simple pursuit-evasion game involving a single agent and single target was used to study the effect of inferring the internal state of the other agent(s). Traditionally, the game has an agent that *pursues* the other agent that tries to *evade* the chasing agent, and vice versa. In this project, the focus is on the behavior of the agent that acts according to the motion of the other agent, denoted as a "target," which has a pre-defined movement models. When the agent is aware of its own role and has full observability of each other's position, the problem can be simply modeled as a Markov decision process (MDP) which can be solved exactly to output an optimal policy [5]. However, when the agent has no a priori knowledge of its own role, it must rely on observation of the target. From this observation, the agent can infer the intention, or the role, of the target. This problem can be modeled as a POMDP and solved using approximate techniques such as QMDP and SARSOP.

In this paper, the mathematical background and the results of solving an MDP model (i.e. agent is aware of its role) and a POMDP model (i.e. agent is unaware of its role) are provided, and performances of solutions in different scenarios are compared and discussed.

II. Background

A. Formulation of Game

The optimal strategies of a single agent in the game of pursuit and evasion were studied. Then, a special case was considered in which an agent has no a priori knowledge of what role it assumes, i.e. agent does not know whether it should pursue after the target or avoid it. In this scenario, an agent must infer the role of the target from the observation of its movement and make an optimal sequence of actions accordingly. For instance, if agent observes that the target continues to move toward it, it is very likely that the target is chasing after the agent, thus agent must evade the target.

The game takes place in a discretized two-dimensional grid world. The game is visualized in Fig. (1), where an agent's position is represented as a green cell, and the target's is shown as an orange cell. The agent's cell also shows the action it takes (shown as "UP") and the posterior probability that the agent is a pursuer when it is unaware of its role (explained further in section II-B2). At each time step, the agent has perfect knowledge of its own and the target's positions, and both agent and target can move one cell at a time.

B. Modeling

When the agent is aware of its role, pursuit and evasion scenarios are modeled as a Markov Decision Process (MDP) defined by the tuple $(S,\ A,\ R,\ T)$. In the MDP model, state space (S) consists of the all possible positions of the agent (α)

Fig. 1. Pursuit-evasion game between an agent (green) and a target (orange) in a grid world

and the target (τ), i.e. $s_t = (\{\alpha_x, \alpha_y\}_t,\ \{\tau_x, \tau_y\}_t) \in S$. The action space ($A$) of the agent consists of the following actions: $a \in (up,\ right,\ down,\ left,\ stay)$. The transition model of the state is defined by $T(s' \mid s, a)$, where s' denotes next state. While the agent's transition is deterministic according to the action a, the transition model of the target depends on its role, and it is explained further in section II-B1. Lastly, the reward $R(s, a)$ is modeled such that there is minor cost of moving a cell, a significant positive reward when the agent catches the target in pursuit game, and a significant negative reward if the agent is caught in the evasion game.

In the scenario where uncertainty in the agent's role is introduced, the state space includes an additional variable ρ_α that denotes the agent's role. θ_p is also used to denote the probability that the agent is a pursuer, i.e. $\theta_p = P(\rho_\alpha = \text{pursuer})$. Because the state at step t, $s_t = (\{\alpha_x, \alpha_y\}_t,\ \{\tau_x, \tau_y\}_t,\ \rho_{\alpha,t})$, is not fully observable, the game is modeled as a Partially Observable Markov Decision Process (POMDP), which is defined by the tuple $(S,\ A,\ R,\ T,\ O,\ Z)$. POMDP model adds an observation space (Z), which in this problem is simply the new position of the target, $\{\tau_x, \tau_y\}_{t+1}$. Then, because the positions of agent and target are fully observable, the observation model $O(o \mid s', a)$, which is the probability of making an observation o given next state s' and action a, is deterministic.

In addition to the tuple defined, the model must specify the transition model of the target and update rule of the state belief b.

1) Target Transition: The transition distribution model of the target $T(\{\tau_x, \tau_y\}' \mid \{\tau_x, \tau_y\}, a_\tau)$ specifies the probabilities associated with each of its five actions (up, right, down, left, stay) at current position. Because the agent updates its belief of ρ_α based on the observation of the target's movement, the target in both scenarios must have distinct enough transition distributions so that the agent's belief can be updated correctly. In this project, the target moves stochastically if the agent is a pursuer. On the other hand, if the agent is an evader, the probability distribution of target's movement varies according to the agent's relative position, as shown in Fig. (2).

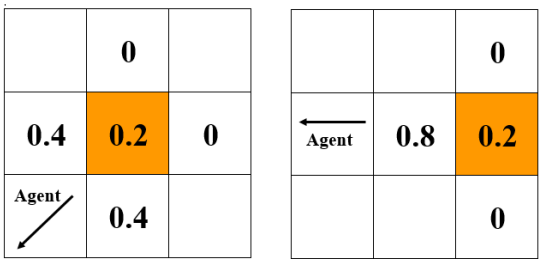

Fig. 2. Probability distribution of target's transition in evasion game

2) Belief Update: In a POMDP model, the state of the system is represented by a probability distribution called belief. In this project, there is only a belief over the agent's role, so when the agent and the target occupy certain cells in the grid world, there are only two states with nonzero belief, $b(s(\rho_\alpha = \text{pursuer})) = \theta_p$, $b(s(\rho_\alpha = \text{evader})) = 1 - \theta_p$. Normally, a belief distribution of a discrete state $b(s)$ can be exactly updated using Eq. (1).

$$b'(s') \propto O(o \mid s', a) \sum_s T(s' \mid s, a) b(s) \tag{1}$$

In this game, the belief that the agent is an evader is increased if the target moves closer to the agent. If the target stays or moves away, the belief that the agent is a pursuer is increased. The exact method in Eq. (1) does not reflect this very well. Therefore, Bayesian parameter learning [5] was used to update the state belief in a prescribed manner. The parameter θ_p = P(ρ_α = pursuer) is described by Beta distribution, which is defined by Eq. (2),

$$\text{Beta}(\theta \mid \pi + 1, \epsilon + 1) = \frac{\Gamma(\pi + \epsilon + 2)}{\Gamma(\pi + 1)\Gamma(\epsilon + 1)} \theta^\pi (1 - \theta)^\epsilon \tag{2}$$

where π, ϵ correspond to the number of times an agent observes the target staying or moving away (i.e. agent is a pursuer) and the target moving towards the agent (i.e. agent is an evader), respectively. Γ refers to a gamma function. Example Beta distributions are illustrated in Fig. (3).
Note that, for instance, if an agent observes the target moving towards the agent 1 time and staying or moving away 5 times, the probability distribution of θ_p is given by Beta(6,2), which, according to Fig. (3), has higher distribution for $\theta_p > 0.5$.

The complete update scheme using Bayesian parameter learning is described in Algorithm 1, where b is a belief distribution that keeps track of π, ϵ, and θ_p in addition to the agent and target positions (α, τ).

C. Model Solutions

When the agent's role is known, the game is modeled as an MDP. The optimal policies $\pi(s)$ of both pursuit and evasion

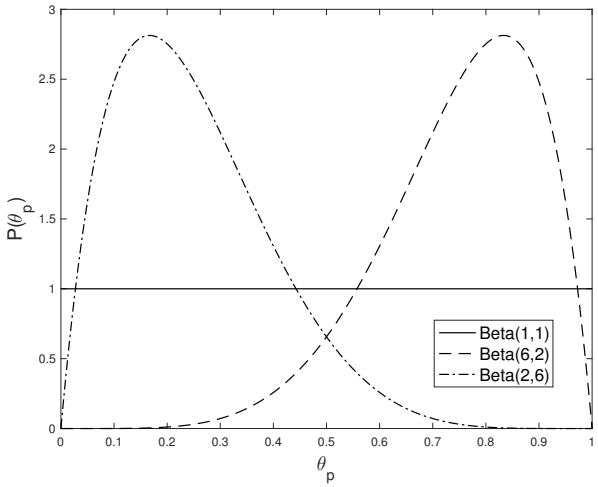

Fig. 3. Beta distributions

Algorithm 1 Bayesian Parameter Learning

function FindDistance(α, τ)
return $\sqrt{(\alpha_x - \tau_x)^2 + (\alpha_y - \tau_y)^2}$
function BeliefUpdate(*b, a, o*)
$d \leftarrow$ FindDistance(α, τ)
$d' \leftarrow$ FindDistance(α, o)
if $d' < d$ **then**
  ϵ++
else
  π++
end if
$P \sim$ Beta(π, ϵ)
$\theta_p \sim$ rand(P)
Update α with action a
$\tau' \leftarrow o$
$b' \leftarrow \alpha'$, τ', π, ϵ, θ_p
return b'

games are solved exactly using the Bellman equation, which is given by Eq. (3).

$$\pi(s) \leftarrow \arg\max_a \Big(R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) U^*(s') \Big) \quad (3)$$

where γ is a discount factor, and $U^*(s)$ is a value associated with an optimal policy at state s.

When the agent's role is unknown the game is modeled as a POMDP. To solve an optimal policy of the POMDP model, a technique called QMDP is used. QMDP computes the alpha vectors α_a, which are piece-wise, linear, convex vectors that form hyperplanes in belief spaces for an action a [5]. The alpha vectors are calculated using Bellman equation similar to Eq. (3),

$$\alpha_a^{(k+1)}(s) = R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) \max_{a'} \alpha_{\alpha'}^{(k)}(s') \quad (4)$$

where $\alpha_a^{(0)}(s)$ is initialized to 0. Then, the optimal policy can be solved using Eq. (5),

$$\pi(s) \leftarrow \arg\max_a \alpha_a^{\mathrm{T}} \mathbf{b} \quad (5)$$

where b is a vector of $b(s)$ for all states.

III. Results and Discussions

In this section, the MDP and POMDP models of pursuit and evasion games are solved using value iteration and QMDP, respectively, and the simulation was run. The results are then evaluated and compared. Table I lists the values of parameters used in the model development and simulations.

TABLE I
MDP and POMDP model parameters

heightParameter	Symbol	Value
Grid world size		15 × 15
Max. number of iterations		100
Discount factor	γ	0.9
Reward of moving a grid		-1
Reward of capture (pursuit game)		100
Reward of capture (evasion game)		-100
Agent initial position	α_0	(1, 2)
Target initial position	τ_0	(10, 8)
Initial belief	$\theta_{p,0}$	0.5

A. Belief Updater Performance

The performance of belief updater outlined in section II-B2 was measured by comparing the beliefs over the agent's role for pursuit and evasion games. For both cases, as noted in Table I, the agent begins the game with equal belief that the agent is either a pursuer or an evader. Figure 4 illustrates how the belief that the agent pursues, θ_p, typically changes over iterations according to the belief update rule described in Algorithm 1. Note that in the pursuit game, the belief θ_p for pursuit game is generally higher than 0.5, indicating that the agent believes it is more likely to be pursuing the target. On the other hand, θ for evasion game tends to stay below 0.3, indicating it is more likely to evade the target. In this particular simulation, the agent successfully captured the target within 27 steps of the pursuit game, while it successfully evaded the target chaser throughout 100 iterations.

B. Comparison of Average Rewards

To gauge the performance of MDP and POMDP models in all scenarios, the simulation was performed a 1,000 times per each scenario, and the average of accrued rewards was computed. Table II lists the average rewards of each scenario with standard deviation.

Notice that, as expected, the average rewards of MDP models for both pursuit and evasion games are higher than those of corresponding POMDP models. Smaller rewards indicate that the agent made more unnecessary movements, since moving one cell has a unit cost. These additional movements are the products of state uncertainty inherent to POMDP models, since the agent must "explore" around the grid world until its

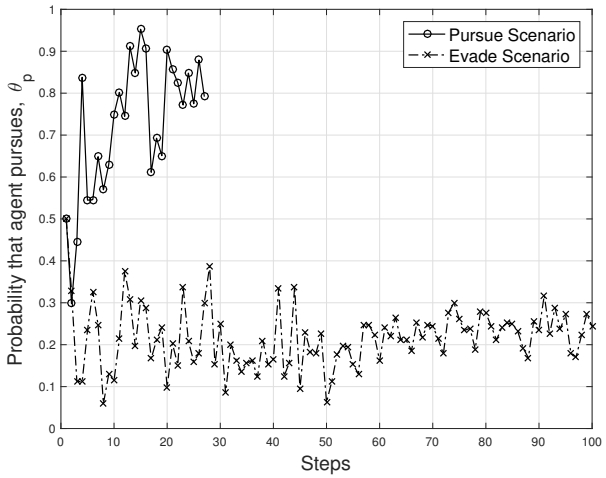

Fig. 4. Belief over timesteps

TABLE II
AVERAGE REWARDS OF DIFFERENT SCENARIOS

heightScenario	Average Reward
Pursuit (MDP)	83.28 $\pm$ 4.44
Pursuit (POMDP)	75.75 $\pm$ 17.06
Evade (MDP)	-66.33 $\pm$ 4.03
Evade (POMDP)	-75.10 $\pm$ 4.82

belief about its own role is biased towards one with enough observations of the target's movements.

Also notice that for pursuit game with unknown agent role, also noted Pursuit (POMDP) in Table II, the average reward has very high standard deviation unlike in other scenarios. This is because the agent does not always catch the target within hundred iterations. Because in a pursuit game the target's movement is stochastic, even though the target *randomly* moves in a direction towards the agent, the agent may perceive this movement as an act of chasing, thus increasing the belief that the agent itself must evade the target. Thus, the prescribed belief updater, which works well most of the time, may have difficulty distinguishing the transitions of targets in pursuit the pursuit game. For Pursuit (POMDP) scenario in Table II, there were 12 incidents out of 1,000 samples in which the agent failed to capture the target.

IV. CONCLUSION

A pursuit-evasion game involving a single agent and a single target was used to study how an agent unaware of its role (pursuer or evader) can make a sequence of optimal decisions based on the observation of the target's movement. The problem was modeled as a POMDP, and its belief of the agent's role was updated using Beta distribution. It was shown that for the given target transition model, the belief updater decently estimates the agent's role. However, according to simulations, the agent failed to capture the target due to incorrect belief updating in 12 out of 1,000 simulations of pursuit game.

The study done in this project can be further enhanced by studying various transition models of the target. In such cases, the value of information [5] associated with observing target movement in POMDP model can be inferred. Theoretically, if target moves stochastically in both pursuit and evasion games, there is no good way of correctly updating the agent's belief of its role. In this case, the value of information should be zero. The value of information will be maximized if there is a clear distinction in how an agent moves when it pursues and evades, i.e. if the target always moves away from the agent in the pursuit game, while it always moves towards the agent in the evasion game.

If the concept in this project is to be expanded to different kinds of games with uncertainty in variables with more than two possibilities, Dirichlet distribution may be used instead of Beta. It may also be useful to try different solvers for POMDP model, such as SARSOP, or both MDP and POMDP models can be solved using online methods such as Monte Carlo tree search [5] and compared.

REFERENCES

[1] R. Siegwart and I. R. Nourbakhsh, "An introduction to autonomous mobile robots," 2nd ed., MIT Press, 2011.
[2] Z. N. Sunberg, C. J. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," 2017 American Control Conference (ACC), Seattle, WA, 2017, pp. 3004–3010.
[3] M. Egorov, M. J. Kochenderfer, and J. J. Uudmae, "Target surveillance in adversarial environments using POMDPs," in AAAI Conference on Artificial Intelligence (AAAI), 2016.
[4] Hoang, T. N., and Low, K. H. 2013. Interactive POMDP Lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In International Joint Conference on Artificial Intelligence (IJCAI), 2298 2305.
[5] M. J. Kochenderfer, "Decision making under uncertainty: theory and application," MIT Press, 2015.