Interplay between Multiagent Games and Generative Adversarial Imitation Learning

Shuijing Liu

10/12/2019

The paper I'm reading today is called

ASYNCHRONOUS MULTI-AGENT GENERATIVE ADVERSARIAL IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning aims to inversely learn a policy from expert demonstrations, which has been extensively studied in the literature for both single-agent setting with Markov decision process (MDP) model, and multi-agent setting with Markov game (MG) model. However, existing approaches for general multi-agent Markov games are not applicable to multi-agent *extensive* Markov games, where agents make asynchronous decisions following a certain order, rather than simultaneous decisions. We propose a novel framework for asynchronous multi-agent generative adversarial imitation learning (AMAGAIL) under general extensive Markov game settings, and the learned expert policies are proven to guarantee subgame perfect equilibrium (SPE), a more general and stronger equilibrium than Nash equilibrium (NE). The experiment results demonstrate that compared to state-of-the-art baselines, our AMAGAIL model can better infer the policy of each expert agent using their demonstration data collected from asynchronous decision-making scenarios (i.e., extensive Markov games).

Roadmap

What is **Generative Adversarial Imitation Learning (GAIL)**

Multiagent GAIL (MAGAIL):

Extend GAIL to multi-agent Markov Games (MG)

Asynchronous MAGAIL (AMAGAIL):

Extend MAGAIL to asynchronous MG, aka EMG

Generative Adversarial Imitation Learning (GAIL)

Remember our old friend GAN…

- We have two models in GAN:
  - A **generative model $\boldsymbol{G}$** that mimics the real data distribution
  - A **discriminative model $\boldsymbol{D}$** that estimates the probability that a sample x came from real data rather than G
    - $D(x)$ is the probability that input x came from real data: $x \sim p_{real}(x)$
    - Similarly, $1 - D(x)$ is the probability that x came from fake data by G: $x \sim p_G(x)$
- We train G and D simultaneously so that
  - D maximizes the probability of assigning correct labels to real data and samples from G
  - G "confuses" D by minimizing this probability
- Formally speaking, D and G play the minimax game with loss function $E(D, G)$:

$$\min_G \max_D E(D, G) = \mathbb{E}_{x \sim p_{real}(x)}[logD(x)] + E_{x \sim p_G(x)}[\log(1 - D(x))]$$

Goodfellow, Ian, et al. "Generative Adversarial Nets." *Advances in neural information processing systems*. 2014.

Use the similar spirit in Imitation Learning…

- In Imitation learning, we are trying to learn a policy π_θ from an expert π_E
  - the "real data" becomes state-action pairs (s, a) sampled from π_E
  - the "fake data" becomes (s, a) sampled from our policy π_θ (or G)

While Discriminator D tries to distinguishes them!

- Continue our minimax game in GAIL:

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \quad (18)$$

6: **end for**

Train D_w to classify whether state-action pairs (s, a) are sampled from π_E or π_θ

Train π_θ with TRPO while maximizing entropy to fool D_w

Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." *Advances in neural information processing systems*. 2016.

Q: Wait… But how to generalize GAIL to Multi-agent Games?

A: Song et al. proposed **Multi-Agent Generative Adversarial Imitation Learning (MAGAIL)** in 2018!

But before we talk about MAGAIL…

Markov Game (MG)

	Single-agent MDP	Multi-agent MG with n players
State Space	$s_t \in S$	$\boldsymbol{s_t} = (s_1, \dots, s_n) \in S^n$
Action Space	$a_t \in A$	$A_1, \dots, A_n$
Transition Function	$T: S \times A \times S \rightarrow [0, 1]$	$T: S \times A_1 \times \cdots \times A_n \times S \rightarrow [0, 1]$
Reward Function	$R: S \times A \rightarrow \mathbb{R}$	For each agent i, $R_i: S \times A_1 \times \cdots \times A_n \rightarrow \mathbb{R}$
Policy	$\pi: S \rightarrow A$	For each agent i: $\pi_i: S^n \rightarrow A_i$
Discount Factor	$0 \leq \gamma < 1$	$0 \leq \gamma < 1$
Initial State Distribution	$p(s_0) \sim \eta$	$p(\boldsymbol{s_0}) \sim \boldsymbol{\eta}$
Objective function	$\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \vert s_0 \sim \eta, a_t \sim \pi(s_t)]$	Each agent i maximizes its own reward: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_i(\boldsymbol{s_t}, a_{i,t}) \vert \boldsymbol{s_0} \sim \boldsymbol{\eta}, a_{i,t} \sim \pi_i(\boldsymbol{s_t})]$

Multi-Agent GAIL

MAGAIL is just GAIL with many D and G

- For each agent i, we need a discriminator D_{w_i} and a policy π_i
- So in each iteration, the D_{w_i} and generator π_i update steps become:

for $u = 0, 1, 2, \dots$ **do**
  Obtain trajectories of size B from π by the process: $s_0 \sim \eta(s), a_t \sim \pi_{\theta_u}(a_t|s_t), s_{t+1} \sim T(s_t|a_t)$.
  Sample state-action pairs from $\mathcal{D}$ with batch size B.
  Denote state-action pairs from π and $\mathcal{D}$ as χ and χ_E.
  for $i = 1, \dots, n$ **do**
    Update ω_i to increase the objective

$$\mathbb{E}_{\chi}[\log D_{\omega_i}(s, a_i)] + \mathbb{E}_{\chi_E}[\log(1 - D_{\omega_i}(s, a_i))]$$

  end for
  for $i = 1, \dots, n$ **do**
    Compute value estimate $V^\star$ and advantage estimate A_i for $(s, a) \in \chi$.
    Update ϕ_i to decrease the objective

$$\mathbb{E}_{\chi}[(V_\phi(s, a_{-i}) - V^\star(s, a_{-i}))^2]$$

    Update θ_i by policy gradient with small step sizes:

$$\mathbb{E}_{\chi}[\nabla_{\theta_i} \pi_{\theta_i}(a_i|o_i) A_i(s, a)]$$

  end for
end for

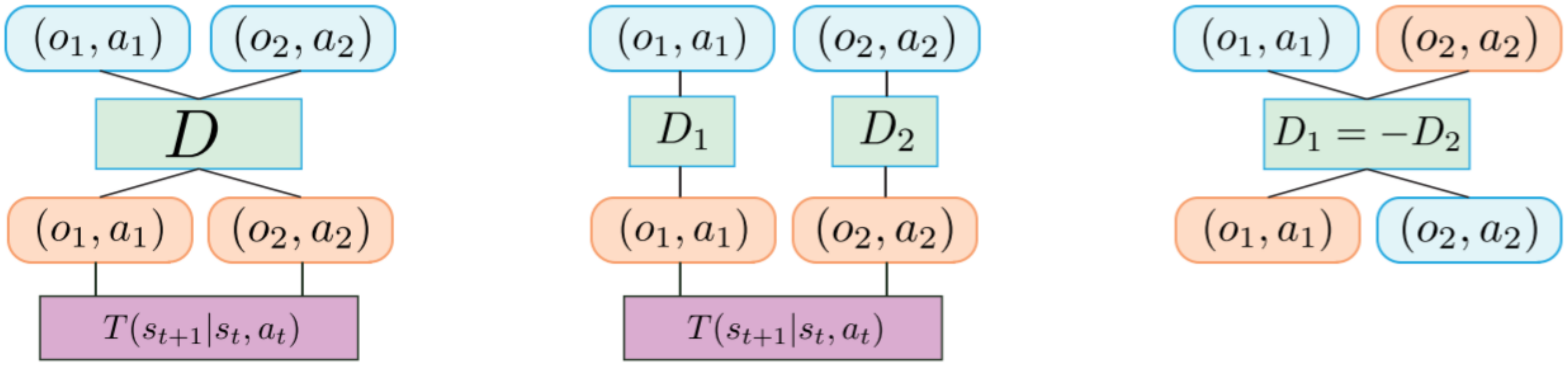
Train each D_{w_i} to classify whether state-action pairs (s, a) are sampled from π_E or π_{θ_i}

Train each π_{θ_i} with TRPO while maximizing entropy to fool D_{w_i}

Song, Jiaming, et al. "Multi-agent generative adversarial imitation learning." *Advances in Neural Information Processing Systems*. 2018.

Wait… If n is large, we'll end up with too many Ds and Gs!

- We can draw relations between $D_1, \dots, D_n$ for different types of MG:



(a) Centralized (Cooperative) (b) Decentralized (Mixed) (c) Zero-sum (Competitive)

$R_1 = R_2 = \cdots = R_n$

Assumes no correlation between $R_1, \dots, R_n$

In a two-player game, $R_1 = -R_2$

Song, Jiaming, et al. "Multi-agent generative adversarial imitation learning." *Advances in Neural Information Processing Systems*. 2018.

Extensive Markov Games (EFG)

EMG: an asynchronous extension of MG

- In **Markov games (MG),** n agents make simultaneous decisions at each timestep, with policies only depending on the current state $\boldsymbol{s_t}$
- In **Extensive Markov games (EMG)** (or Extensive form games (EFG) in the paper), n agents make asynchronous decisions, with policies conditioned on the entire history of the game

Anonymous. "Multi-agent generative adversarial imitation learning." In submission to International Conference on Learning Representations, ICLR 2020.

	MG with n players	EMG with n players
State Space	$\boldsymbol{s_t} = (s_1, \dots, s_n) \in S^n$	$\boldsymbol{s_t} = (s_1, \dots, s_n) \in S^n$
Action Space	$A_1, \dots, A_n$	$A_1' = A_1 \cup \{\phi\}, \dots, A_n' = A_n \cup \{\phi\}$, where ϕ denotes no participation
Participation Vector	$None$	$\boldsymbol{I_t} = [I_{1,t}, \dots, I_{N,t}]$ indicates active vs inactive agents at time t $h_{t-1} = [\boldsymbol{I_0}, \dots, \boldsymbol{I_{t-1}}]$ is the participation history from time 0 to $t-1$
Player function	$None$	$Y(i \mid h_{t-1})$ describes the probability of agent i to make an action at time t, given the participation history h_{t-1}
Transition Function	$T: S \times A_1 \times \cdots \times A_n \times S \to [0, 1]$	$T: S \times A_1 \cup \{\phi\} \times \cdots \times A_n \cup \{\phi\} \times S \to [0, 1]$
Reward Function	For each agent $i, R_i: S \times A_1 \times \cdots \times A_n \to \mathbb{R}$	For each agent $i, R_i: S \times A_1 \times \cdots \times A_n \to \mathbb{R}$
Policy	For each agent i: $\pi_i: S^n \to A_i$	For each agent i: $\pi_i: S^n \to A_i \cup \{\phi\}$
Discount Factor	$0 \leq \gamma < 1$	$0 \leq \gamma < 1$
Initial State Distribution	$p(\boldsymbol{s_0}) \sim \boldsymbol{\eta}$	$p(\boldsymbol{s_0}) \sim \boldsymbol{\eta}$
Objective function	Each agent i maximizes its own reward: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_i(\boldsymbol{s_t}, a_{i,t}) \mid \boldsymbol{s_0} \sim \boldsymbol{\eta}, a_{i,t} \sim \pi_i(\boldsymbol{s_t})]$	Each agent i maximizes its own reward: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_i(\boldsymbol{s_t}, a_{i,t}) \mid \boldsymbol{s_0} \sim \boldsymbol{\eta}, a_{i,t} \sim \pi_i(\boldsymbol{s_t})]$

Asynchronous MAGAIL

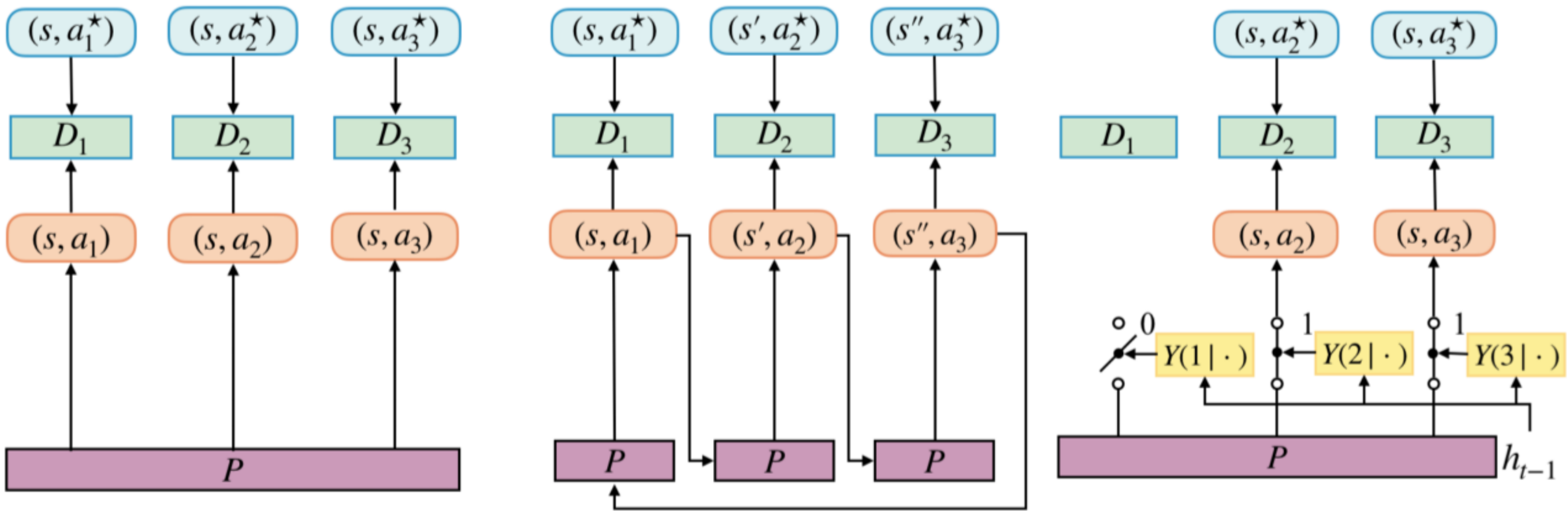
AMAGAIL algorithm is nearly identical to MAGAIL

for u = 0,1,2,... **do**
  Generate state-action pairs of batch size B from π_u through the process: $s_0 \sim \eta(s)$, $\boldsymbol{I}_0 \sim \zeta, \boldsymbol{I}_t \sim Y, \boldsymbol{a} \sim \boldsymbol{\pi}^*(\cdot|s_t), s_{t+1} \sim P(s_{t+1}|s_t, \boldsymbol{a})$; denote the generated state-action pairs as $\mathcal{X}$.

  Sample state-action pairs from $\mathcal{Z}$ of batch size B; denote the demonstrated state-action pairs as $\mathcal{X}_E$.
  for i = 1,...,N **do**
    Update w_i to increase the objective
    $\mathbb{E}_{\mathcal{X}}[\log D_{w_i}(s, a_i)] + \mathbb{E}_{\mathcal{X}_E}[\log(1 - D_{w_i}(s, a_i))]$
  end for
  for i = 1,...,N **do**
    Compute value estimate V^* and advantage estimate A_i for $(s, a) \in \mathcal{X}$.
    Update ϕ_i to decrease the objective
    $\mathbb{E}_{\mathcal{X}}[(V_\phi(s, a_{-i}) - V^*(s, a_{-i}))^2]$
    Update θ_i by policy gradient with the setting step sizes:
    $\mathbb{E}_{\mathcal{X}}[\nabla_{\theta_i} \pi_{\theta_i}(a_i|s_i) A_i(s, a)]$
  end for
end for

Anonymous. "Multi-agent generative adversarial imitation learning." In submission to International Conference on Learning Representations, ICLR 2020.

AMAGAIL with 3 different participation rules



(a) Synchronous participation

(b) Deterministic participation

(c) Stochastic participation

The player function $Y(i|h_{t-1}) = 1$ for all agents at all timesteps, in which case EMG becomes MG.

All agents take turns to make actions with a fixed order.

All agents have stochastic player functions (yellow boxes)

Anonymous. "Multi-agent generative adversarial imitation learning." In submission to International Conference on Learning Representations, ICLR 2020.

That’s it! Thank you!