Statistical Techniques in Robotics (16-831, S21) Lecture #25 (Wednesday, May 5)

Passive Imitation Learning & Interactive Imitation Learning

Lecturer: Kris Kitani *Scribes: Xuhua Huang, Xinjie Yao*

1 Review

This is the final lecture of this semester. First of all, great thanks to **Kris Kitani** and **Xingyu Lin** for such a great learning experience! In the last lecture, we get started on a new topic called Imitation Learning (IL), which is a method aiming to learn and develop own skills by observing skills performed by others. As a high-level summary, we have:

	Passive IL	Active IL	Interactive IL
Demonstrations $\mathcal{D}$	yes	yes	optional
Environment $\mathcal{E}$	no	yes	yes
Oracle π	no	no	yes
Dynamics $\mathcal{T}$	no	optional	optional
Reward $\mathcal{R}$	no	optional	optional

Table 1: Settings for three types of Imitation Learning

In terms of feedback form,

- Passive IL: sampled, sequential and instructive
- Active IL: sampled, sequential and evaluative
- Interactive IL: sampled, sequential and evaluative + instructive

In this write-up, we will first give a brief review of Generative Adversarial Imitation Learning (GAIL) learned from last lecture, which is also one of the most standard and popular methods in Active IL field. Then, we will have a comparison among these three Imitation Learning types in the Summary section, and move on to introduce Passive IL. After reviewing the problems of Passive IL, we will introduce two representative Interactive IL methods, namely DAgger and AggreVaTe.

1.1 Generative Adversarial Imitation Learning (GAIL)

In GAIL, the generator is the policy network and the discriminator is the reward function network. The policy tries to generate expert-like trajectories and gets better to get more reward. While the scoring function tries to detect fake trajectories and gets better at identifying fake ones.
Given the true data $\mathcal{D}^*$ and the fake data $\mathcal{D}_\theta$, the discriminator learns to optimize

$$\max_{\phi} \left\{ \sum_{s,a \sim \mathcal{D}_\theta} [D_\phi(s,a)] + \sum_{s,a \sim \mathcal{D}^*} [1 - D_\phi(s,a)] \right\}$$

Given the discriminator, the generator learns to optimize

$$\min_{\theta} \left\{ \sum_{s,a \sim \mathcal{D}_\theta} [D_\phi(s,a)] \right\}$$

By putting two objectives together, GAIL learns the generator and discriminator at the same time by optimizing

$$\min_{\theta} \max_{\phi} \left\{ \sum_{s,a \sim \mathcal{D}_\theta} \ln [D_\phi(s,a)] + \sum_{s,a \sim \mathcal{D}^*} \ln [1 - D_\phi(s,a)] \right\}$$

Then we have the GAIL algorithm as Algo 1.

Algorithm 1 Simplified GAIL

$\mathcal{D}^* = \{\zeta_n^*\}_{m=1}^M$ where $\zeta_m^* = \{s^{(0)}, a^{(0)}, \ldots, s^{(T_m)}\} \sim \mathcal{E}|\pi^*$
for $k = 1, \ldots K$ **do**
  $\mathcal{D}_\theta = \{\zeta_n\}_{n=1}^N$ where $\zeta_n = \{s^{(0)}, a^{(0)}, \ldots, s^{(T_n)}\} \sim \mathcal{E}|\pi_\theta$
  $\phi = \phi + \mathbb{E}_{\mathcal{D}_\theta}[\nabla_\phi \ln D_\phi(a,s)] + \mathbb{E}_{\mathcal{D}^*}[\nabla_\phi \ln\{1 - D_\phi(a,s)\}]$
  $\theta = \theta - \mathbb{E}_{\mathcal{D}_\theta}[\nabla_\theta \ln \pi_\theta(a|s) Q(a,s)]$
end for
return $\hat{\pi}$

2 Summary

This section will first introduce the definition of Passive IL, and analyze its drawbacks. Then to solve the issues of Passive IL, we will introduce Interactive IL and two representative methods of it. In the end, we will wrap up the difference among Interactive IL, Passive IL, Active IL, Inverse Reinforcement Learning (IRL) and traditional Reinforcement Learning (RL).

2.1 Passive Imitation Learning

As mentioned in Table 1, Passive IL only has access to the experience *offline*. It can also be called Supervised Learning or Behavior Cloning. In our normal settings, experience can be the expert demonstrations which are a set for state-action pairs. The goal is to learn a policy ONLY based on these expert demonstrations.

However, one drawback of Passive IL is caused by *Covariate Shift*. *Covariate Shift* usually happens in sequential feedback process, where the expert demonstrations do not cover some abnormal states and the recovery actions to deal with those states, while these unseen states will happen in test environment. If the agent keeps accumulating small errors with many steps, it will cause compounding error and finally push the agent to unseen state, then fail.

Therefore, though Passive IL can work very well if you have lots of data covering all possible states, if we only have limited expert demos, it will fail because it cannot model long-term behavior well. Things can become much better when we have more data, especially those data for unseen states. This leads us to the next type of Imitation Learning method called Interactive Imitation Learning.

2.2 Interactive Imitation Learning

The main difference between Interactive IL and Passive IL is, for Interactive IL, we now can interact with the environment and query the oracle / expert feedback during the learning process. Below we will introduce two popular algorithms built on this idea.

2.3 DAgger

DAgger[2] stands for Data Aggregation, and in general, it's described in Figure 1. The learner firstly executes the current policy and queries expert. The expert provides new data using the expert policy which are aggregated together with all previous data. Then in an supervised learning manner, the learner picks up the new optimal policy and repeats this procedure.

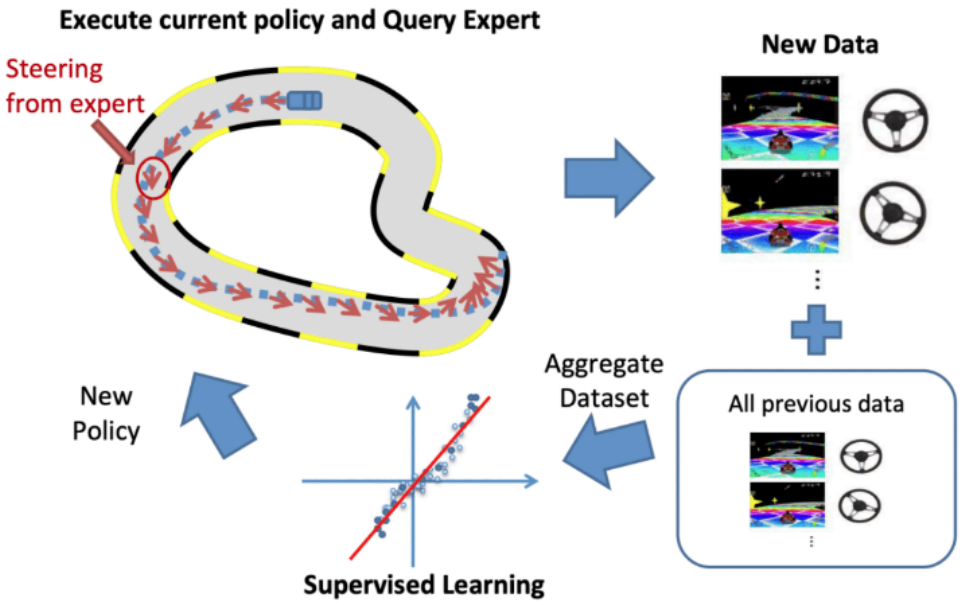

Figure 1: DAgger illustration

The algorithm is presented in Algo 2.

Algorithm 2 DAgger $(\beta, \pi^*, \Pi, \mathcal{E})$

for $k = 1, \cdots, K$ **do**
  $\pi_k = \beta\pi^* + (1-\beta)\hat{\pi}_k$ ▷ mixture expert policy and hindsight optimal policy
  $\left\{s^{(t)}, a^{(t)}\right\}_{t=1}^{T} \sim \mathcal{E} \mid \pi_k$ ▷ off-policy trajectory sample
  $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{s^{(t_k)}\right\}$ ▷ data aggregation
  $\hat{\pi}_{k+1} = \arg\min_{\pi \in \Pi} \sum_{d=1}^{|\mathcal{D}|} f\left(\pi\left(s^{(d)}\right), \pi^*\left(s^{(d)}\right)\right)$ ▷ hindsight optimal (leader) policy
end for
return $\hat{\pi}$

Recall that online convex optimization is a generalization of online learning when the loss function is convex. We learn Follow-the-Leader algorithm as a general framework for online learning. If we can construct a convex loss function of the expert policy and the learner policy, Interactive Imitation Learning is Online Convex Optimization and DAgger is just Follow-the-Leader algorithm with the no-regret guarantee.

Here we summarize the pros and cons of DAgger as follows.
Pros:

- Trajectory sampling implicitly accounts for sequential dependence of actions (addresses domain shift problem)
- Reduction of IL to online learning results in no-regret bounds

Cons:

- Solving for the optimal policy for each iteration can be a costly optimization
- All mistakes are treated equally. No explicit reasoning about sequential dependence (e.g., future payoff, value function or cost-to-go)

2.4 AggreVaTe

The main improvement of Aggregate Values to Imitate (AggreVaTe[1]) compared with DAgger is it starts to take sequential dependence (e.g. future payoff) into consideration, by assigning more penalty to more fatal mistake. It leverages cost-to-go information.

The algorithm is presented in Algo 3.

Algorithm 3 AGGREVATE $(\beta, \pi^*, \Pi, \mathcal{E})$

1: **for** $k = 1, \cdots, K$ **do**
2: $\quad \pi_k = \beta\pi^* + (1-\beta)\hat{\pi}_k$ ▷ mixture expert policy and optimal policy
3: $\quad t_k \sim [T]$ ▷ sample switch point
4: $\quad \left\{s^{(i)}\right\}_{i=0}^{t_k} \sim \mathcal{E} \mid \pi_k$ ▷ roll out with expert
5: $\quad$ RECEIVE $\boldsymbol{Q}\left(s^{(t_k)}\right)$ ▷ roll out and get the Q value
6: $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \left\{s^{(t_k)}, \boldsymbol{Q}\left(s^{(t_k)}\right)\right\}$ ▷ record roll out Q for each action
7: $\quad \hat{\pi}_{k+1} = \arg\min_\pi \sum_{d=1}^{|\mathcal{D}|} \sum_a \pi\left(a \mid s_d\right) Q^*\left(s_d, a\right)$ ▷ pick cost sensitive policy
8: **end for**
9: **return** $\hat{\pi}$

AggreVaTe will stop at a random time step t and start to take exploratory action, then follow the expert policy until we reach the end and get the Q value. Its cost sensitive policy will learn to minimize the probability of taking the action leading to low Q value by line 7. Notice that Q can be empirically calculated from Monte Carlo or TD(λ) prediction.

In conclusion, the AggreVaTe will try to minize the **cost** of mistakes (vs. **number** of mistakes for DAgger), by asking expert the **consequence** of mistakes (vs. **occurence** of a mistake for DAgger).

2.5 Comparisons

In this section we will use Figure 2 and Table 2 to illustrate the major differences among Interactive IL, Passive IL, Active IL, Inverse Reinforcement Learning (IRL) and traditional Reinforcement Learning (RL).

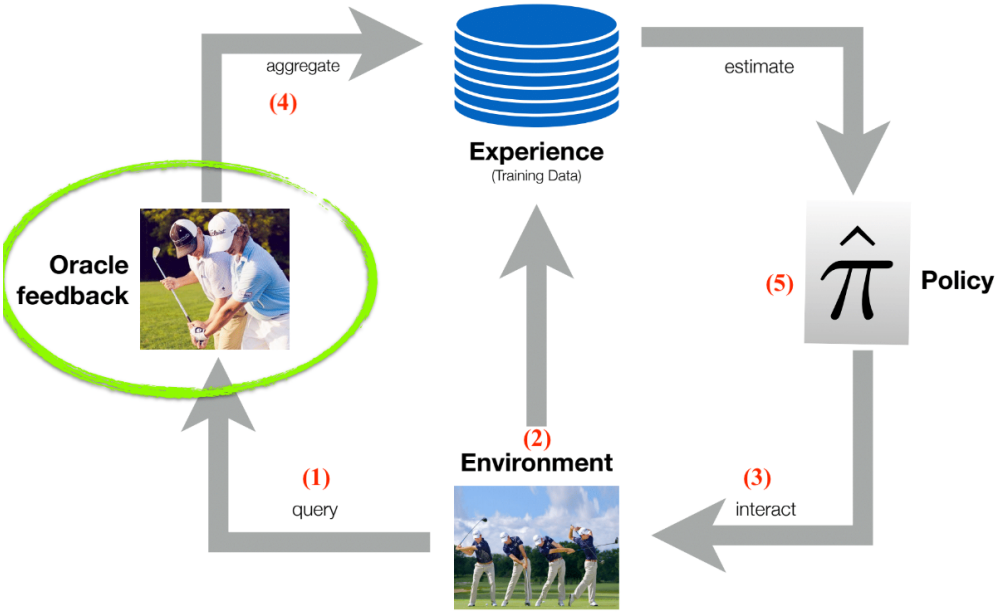

Figure 2: Graph used for comparison

	Passive IL	Active IL	Interactive IL	Inverse RL	RL
(1)	no	no	yes	no	no
(2)	no	yes	yes	yes	yes
(3)	no	yes	yes	yes	yes
(4)	yes	yes	yes	yes	no
(5)	only policy $\hat{\pi}$	only policy $\hat{\pi}$	only policy $\hat{\pi}$	policy $\hat{\pi}$ + reward $\hat{\mathbf{R}}$	only policy $\hat{\pi}$

Table 2: Difference among 5 algorithms. *yes* means the algorithm have access.

2.6 Conclusion

When it comes to sequential process which requires sequential decisions, things will become much more complicated. But with online learning and no-regret analyses, we can introduce different Imitation Learning algorithms to solve this difficult problem. Due to the design of Imitation Learning, we can avoid global explorations, which is much more efficient or even realizable when state/action space are infinite. Combining Imitation and Reinforcement can also train an agent outperforming our experts.

References

[1] S. Ross and J. A. Bagnell. Reinforcement and imitation learning via interactive no-regret learning, 2014.

[2] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.