Imitation Learning: Behavior Cloning

Alina Vereshchaka

CSE4/510 Reinforcement Learning
Fall 2019

avereshc@buffalo.edu

October 10, 2019

*Slides are adopted from Berkley Deep RL course CS294-112 & Deep RL and Control, CMU 10703
Reinforcement Learning: Learning policies guided by **sparse** rewards, e.g., win the game.

- **Good**: simple, cheap form of supervision
- **Bad**: High sample complexity

Where is it successful so far?

- In simulation, where we can afford a lot of trials, easy to parallelize
- Not in robotic systems:
  - action execution takes long
  - we cannot afford to fail
  - safety concerns

Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain, Silver et al. 2010
Ideally we want **dense in time** rewards to closely guide the agent closely along the way.

Who will supply those shaped rewards?

1. **We will manually design them**: “cost function design by hand remains one of the 'black arts' of mobile robotics, and has been applied to untold numbers of robotic systems”

2. **We will learn them from demonstrations**: “rather than having a human expert tune a system to achieve desired behavior, the expert can demonstrate desired behavior and the robot can tune itself to match the demonstration”

Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain, Silver et al. 2010
Learning from demonstrations a.k.a. Imitation Learning:
Supervision through an expert (teacher) that provides a set of demonstration trajectories: sequences of states and actions.

Imitation learning is useful when it is easier for the expert to demonstrate the desired behavior rather than:
   a) coming up with a reward function that would generate such behavior,
   b) coding up with the desired policy directly.
and the sample complexity is manageable
Two broad approaches:

- **Direct**: Supervised training of policy (mapping states to actions) using the demonstration trajectories as ground-truth (a.k.a. behavior cloning)

- **Indirect**: Learn the unknown reward function/goal of the teacher, and derive the policy from these, a.k.a. Inverse Reinforcement Learning
Supervised training
- Behavior Cloning: Imitation learning as supervised learning
- Compounding errors
- Demonstration augmentation techniques
- DAGGER

Inverse reinforcement learning
- Feature matching
- Max margin planning
- Maximum entropy IRL
Terminology & Notations

$o_t$ - state
$π_θ(a_t|o_t)$ - policy
$a_t$ - action
$π_θ(a_t|o_t)$ - policy (fully observed)
$π_θ(a_t|s_t)$ - policy (fully observed)
$s_t$ - state
$o_t$ - observation
$a_t$ - action

Markov property independent of $s_{t-1}$
Terminology & Notations

- $o_t$ – observation
- $u_t$ – action
- $\pi(\cdot | o_t)$ – policy

A bit of history...
- $x_t$ – state
- $u_t$ – action
- $s_t$ – state
- $a_t$ – action

Lev Pontryagin

Richard Bellman
Imitation Learning

**o_t** \[ \pi_{\theta}(a_t|o_t) \] **a_t**

**o_t** \[ \pi_{\theta}(a_t|o_t) \]

**behavior cloning**

Images: Bojarski et al. ‘16, NVIDIA
Does it work?
Data Distribution Mismatch

\[ p_{\pi^*}(o_t) \neq p_{\pi_\theta}(o_t) \]
Behavioral Cloning

- No matter how good it, the policy will make a mistake
- Small errors compound over time
- New states will be completely new to the agent, that wasn’t in the training set
- Eventually it may fail
- Decisions are purposeful, in supervised learning we don’t have a goal or planning problem
Does it work? Yes!

Video: Bojarski et al. '16, NVIDIA
Can we make it work more often?

stability
Can we make it work more often?

\[
p_{\pi_{\theta}}(o_t) \quad \text{training trajectory}
\]

\[
p_{\pi_{\theta}}(o_t) \quad \pi_{\theta} \text{ expected trajectory}
\]

\[
\pi_{\theta}(a_t | o_t)
\]

can we make \( p_{\text{data}}(o_t) = p_{\pi_{\theta}}(o_t) \)?