# 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

# So far

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

Offroad navigation

Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain, Silver et al. 2010

# Reward Shaping

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 demonstration

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 managable

# Imitation Learning

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

# Outline

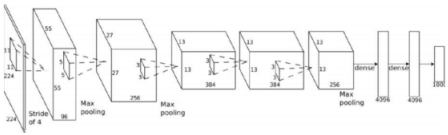
**Supervised training**

- Behavior Cloning: Imitation learning as supervised learning
- Compounding errors
- Demonstration augmentation techniques
- DAGGER

**Inverse reinforcement learning**

- Feature matching
- Max margin planning
- Maximum entropy IRL

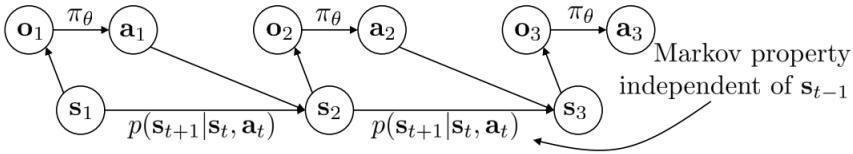
# Terminology & Notations



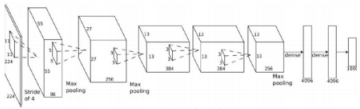

$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



# Terminology & Notations



$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy

a bit of history...

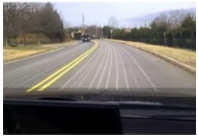
$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action
управление

Lev Pontryagin

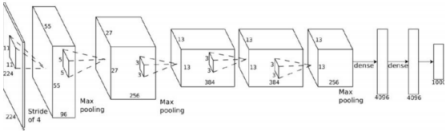
$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Richard Bellman

# Imitation Learning



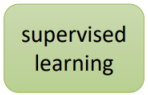
$\mathbf{o}_t$




$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{a}_t$

$\mathbf{o}_t$
$\mathbf{a}_t$

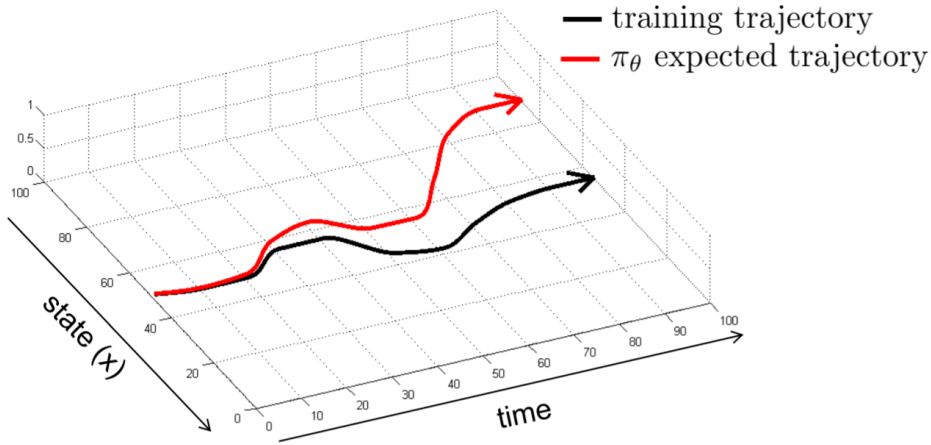




$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

behavior cloning

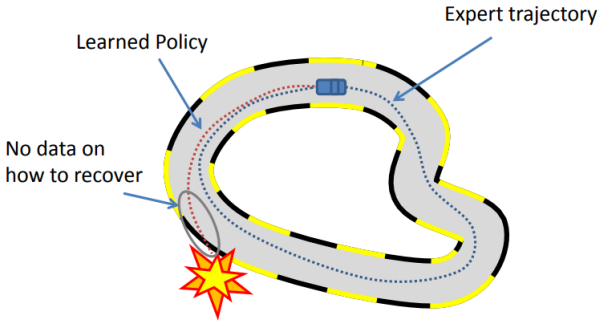
Images: Bojarski et al. '16, NVIDIA

# Does it work?

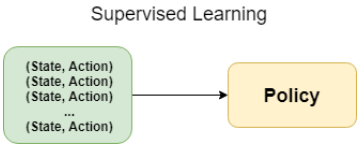

training trajectory
$\pi_\theta$ expected trajectory
1
0.5
0
100
80
60
40
20
0
state (x)
0
10
20
30
40
50
60
70
80
90
100
time

# Data Distribution Mismatch

$$p_{\pi^*}(o_t) \neq p_{\pi_\theta}(o_t)$$



# Behavioral Cloning

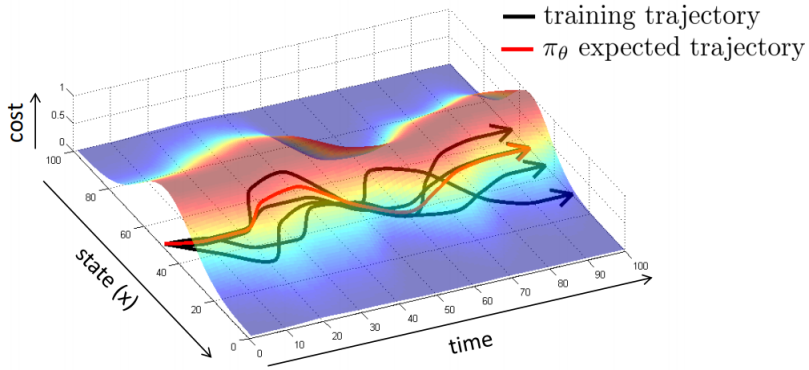



- No matter how good it, the policy will make a mistake
- Small errors compound over time
- New states will be completely new to the agent, that wasn't in the training set
- Eventually it may fail
- Decisions are purposeful, in supervised learning we don't have a goal or planning problem

Does it work? Yes!

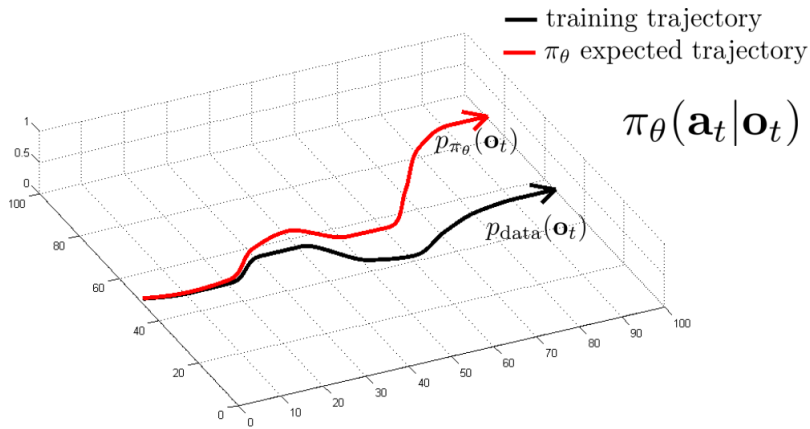
Video: Bojarski et al. '16, NVIDIA

# Can we make it work more often?




stability

# Can we make it work more often?




can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?