CSE 571
Imitation Learning and Policy gradient

Dieter Fox

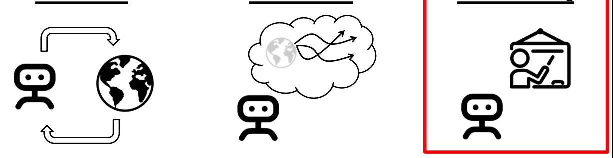
Slides borrowed from many sources – Abhishek Gupta, Liyiming Ke, Sergey Levine

1

How can we learn policies?

$$\max_\theta \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$$

Model-free RL | Model-based RL | Imitation Learning



2

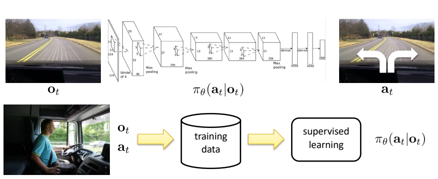
Idea 1: Imitation Learning via Behavior Cloning

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator

$$\arg\max_\theta \mathbb{E}_{(s^*, a^*) \sim \mathcal{D}} [\log \pi_\theta(a^* | s^*)]$$

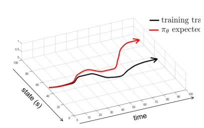
Idea: Treat imitation learning as a supervised learning problem!



3

So does behavior cloning really work?

- Imitation Learning ≠ Supervised Learning

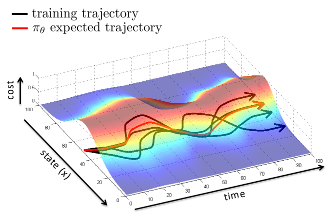


$$\arg\max_\theta \mathbb{E}_{(s^*, a^*) \sim \mathcal{D}} [\log \pi_\theta(a^* | s^*)] \qquad \mathbb{E}_{(s,a) \sim \rho(\pi)} [1(a = a^*)]$$

Not the same!

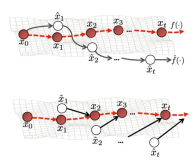
4

What is the general principle?



stability

Corrective labels that bring you back to the data



5

Concrete Instantation: DAgger

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?
idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$
how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$
but need labels $\mathbf{a}_t$!

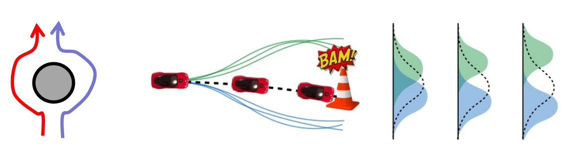
1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

6

Why might we fail to fit the expert?

Multimodal behavior.. amongst other reasons

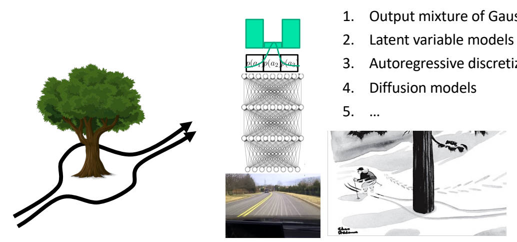


Not a matter of network size! It's about distributional expressivity

7

Why might we fail to fit the expert?

Multimodal behavior → use more **expressive** probability distributions

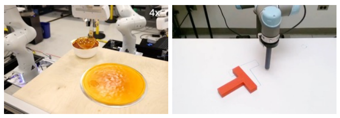


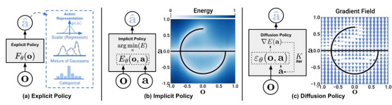
1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

8

Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …



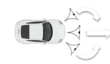


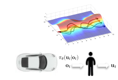
9

Perspectives on Imitation – don't believe everything you see online

$\mathbf{o}_t$, $\mathbf{a}_t$ → training data → supervised learning → $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

- Pros:
  - Easy to use, no additional infra
  - Can sometimes be unreasonably effective
- Cons:
  - Challenges of compounding error, multimodality
  - Doesn't really generalize
  - Very expensive in terms of data collection!



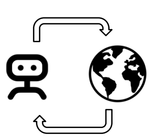


10

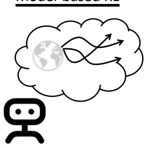
How can we learn policies?

$$\max_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$$

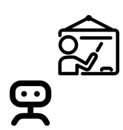
Model-free RL



Model-based RL



Imitation Learning



11

What if we just performed gradient ascent?

$$\max_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$$

$$= \int p_\theta(\tau) R(\tau) d\tau$$

Standard gradient descent (supervised learning)

$$\nabla_\theta \mathbb{E}_{x\sim g(x)}\left[f_\theta(x)\right]$$

REINFORCE gradient descent (RL)

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta(x)}\left[f(x)\right]$$

12

Taking the gradient of return

16

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \log p_\theta(\tau) \sum_{t=0}^{T} r(s_t, a_t) \right]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\substack{s_0 \sim p(s_0) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t) \\ a_t \sim \pi(a_t|s_t)}} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{t'=0}^{T} r(s_t, a_t) \right]$$

$$\approx \frac{1}{N} \sum_{i=0}^{N} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \sum_{t'=0}^{T} r(s_{t'}^i, a_{t'}^i)$$ (approximating using samples)

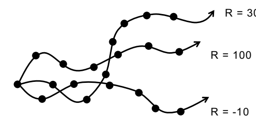
13

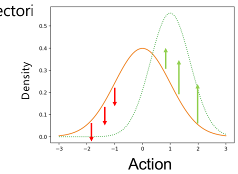
What does this mean?

17

$$\nabla_\theta J(\theta) = \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) d\tau \approx \frac{1}{N} \sum_{i=0}^{N} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) \sum_{t'=0}^{T} r(s_{t'}^i, a_{t'}^i)$$

Increase the likelihood of actions in high return trajectori



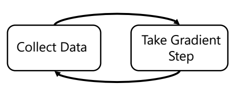


14

Resulting Algorithm (REINFORCE)

18

$$\nabla_\theta J(\theta) = \int p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) d\tau$$

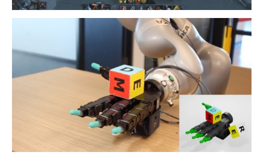


REINFORCE algorithm:

On-policy →
1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ (run it on the robot)
2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i) \right) \left(\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i) \right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

15

Policy Gradient in Action



16