Imitation Learning

CISC 7404 - Decision Making

Steven Morad

University of Macau

Admin 2
Final Projects 6
Finish DPG 12
Imitation Learning 13
Behavioral Cloning 19
Coding 38
Applications 47

Admin

Admin

TAs currently grading homework 1:

Admin

TAs currently grading homework 1:

- Grades will be released over the next week on Moodle

Admin

TAs currently grading homework 1:

- Grades will be released over the next week on Moodle
- All grades should be released by Wednesday

Admin

TAs currently grading homework 1:

- Grades will be released over the next week on Moodle
- All grades should be released by Wednesday

Homework 2 should be graded by the following Wednesday (April 16)

Admin

TAs currently grading homework 1:

- Grades will be released over the next week on Moodle
- All grades should be released by Wednesday

Homework 2 should be graded by the following Wednesday (April 16)

Admin

I am currently grading quiz 2, 40% done, score released soon

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Highly multimodal distribution

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Highly multimodal distribution

- Modes at 20%, 45%, and 80%

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Highly multimodal distribution

- Modes at 20%, 45%, and 80%
  - I drop lowest quiz score, if you did poorly study more next time

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Highly multimodal distribution

- Modes at 20%, 45%, and 80%
  - I drop lowest quiz score, if you did poorly study more next time
- Some students received perfect scores, good job!

Admin

I am currently grading quiz 2, 40% done, score released soon

- I want scoring to be consistent for all students
  - I grade exams myself (not TA), so grading is slow

Highly multimodal distribution

- Modes at 20%, 45%, and 80%
  - I drop lowest quiz score, if you did poorly study more next time
- Some students received perfect scores, good job!
  - He Enhao
  - Leonard Hangqin Zhuang
  - Qiao Yulin
  - Fu Zexin

Admin

We must have one more quiz

Admin

We must have one more quiz

Current options:

Admin

We must have one more quiz

Current options:

- 17 April
- 24 April (last lecture)

Admin

We must have one more quiz

Current options:

- 17 April
- 24 April (last lecture)

Leaning towards 17 April so we can spend all of 24 on LLMs

Final Projects

Final Projects

Project plans turned in

Final Projects

Project plans turned in

- Legend of Zelda
- Super Mario (7x)
- LLM finetuning
- Honor of Kings (3x)
- Federated learning
- Sudoku
- Snake (2x)
- Health AI system
- Tetris
- PacMan (2x)
- Navigation
- StarCraft II
- Getting Over It
- Gymnax
- 2048
- Pokemon Double Battle

Final Projects

Most plans look good

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

I asked for projects that can improve the world

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

I asked for projects that can improve the world

No human should ever have to play *Getting Over It*

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

I asked for projects that can improve the world

No human should ever have to play *Getting Over It*

Training an agent to play instead can improve many human lives

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

I asked for projects that can improve the world

No human should ever have to play *Getting Over It*

Training an agent to play instead can improve many human lives

At university, my friend skipped school for a week to win this game

Final Projects

Most plans look good

I left some comments/suggestions in Moodle

I want to highlight the most creative project plan (in my opinion)

Getting Over It by Bennet Foddy Group: Getting Over It

I asked for projects that can improve the world

No human should ever have to play *Getting Over It*

Training an agent to play instead can improve many human lives

At university, my friend skipped school for a week to win this game

https://youtu.be/8qGCleYV4cw?si=ynB0Idg5-TdAiAh9&t=74

Final Projects

Dr. Bennet Foddy (game author) teaches at NYU

Final Projects

Dr. Bennet Foddy (game author) teaches at NYU

His research focus is on addiction and reward/punishment

Final Projects

Dr. Bennet Foddy (game author) teaches at NYU

His research focus is on addiction and reward/punishment

> This chapter compares … data on different addictions … from drug addiction to binge-eating disorders, gambling, and videogame addiction. … Based on these data, it is argued that **there is a hazard inherent in any rewarding operant behavior, no matter how apparently benign: that we may become genuinely "addicted" to any behavior that provides operant reward**. With this in mind, addiction is rightly seen as a possibility for any human being, not a product of the particular pharmacological or technological properties of any one particular substance or behavior.

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

$$\text{study} + \gamma \text{ exercise} + \gamma^2 \text{ sleep} + \ldots = 10$$

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

$$\text{study} + \gamma \text{ exercise} + \gamma^2 \text{ sleep} + \ldots = 10$$

$$\text{no money} + \gamma \text{ no sleep} + \gamma^2 \text{ pain} + \ldots + \underbrace{\gamma^n \text{ addiction pleasure}}_{\text{Too powerful}} = 100$$

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

$$\text{study} + \gamma \text{ exercise} + \gamma^2 \text{ sleep} + ... = 10$$

$$\text{no money} + \gamma \text{ no sleep} + \gamma^2 \text{ pain} + ... + \underbrace{\gamma^n \text{ addiction pleasure}}_{\text{Too powerful}} = 100$$

If you have no addiction, then you are not following an optimal policy

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

$$\text{study} + \gamma \text{ exercise} + \gamma^2 \text{ sleep} + \ldots = 10$$

$$\text{no money} + \gamma \text{ no sleep} + \gamma^2 \text{ pain} + \ldots + \underbrace{\gamma^n \text{ addiction pleasure}}_{\text{Too powerful}} = 100$$

If you have no addiction, then you are not following an optimal policy

Optimal policy in humans **will** lead to bad behavior

Final Projects

We become addicted (gambling, drugs, etc) because of reward!

Drugs/gambling/etc reward is so powerful it overwhelms the rewards of normal life

$$\text{study} + \gamma \text{ exercise} + \gamma^2 \text{ sleep} + \ldots = 10$$

$$\text{no money} + \gamma \text{ no sleep} + \gamma^2 \text{ pain} + \ldots + \underbrace{\gamma^n \text{ addiction pleasure}}_{\text{Too powerful}} = 100$$

If you have no addiction, then you are not following an optimal policy

Optimal policy in humans **will** lead to bad behavior

- Addict behavior policy maximizes the return!

Final Projects

Getting Over It represents this addiction

Final Projects

Getting Over It represents this addiction

$$\text{fall} + \gamma \text{ anger} + \ldots + \gamma^n \text{ win game} > \text{study} + \gamma \text{ exercise} + \ldots$$

Final Projects

Getting Over It represents this addiction

$$\text{fall} + \gamma \text{ anger} + ... + \gamma^n \text{ win game} > \text{study} + \gamma \text{ exercise} + ...$$

It is always interesting to consider our own reward functions

Final Projects

Getting Over It represents this addiction

$$\text{fall} + \gamma \text{ anger} + \ldots + \gamma^n \text{ win game} > \text{study} + \gamma \text{ exercise} + \ldots$$

It is always interesting to consider our own reward functions

Humans are born with reward functions and cannot choose them

Final Projects

Getting Over It represents this addiction

$$\text{fall} + \gamma \text{ anger} + ... + \gamma^n \text{ win game} > \text{study} + \gamma \text{ exercise} + ...$$

It is always interesting to consider our own reward functions

Humans are born with reward functions and cannot choose them

We can select good reward functions for artificial agents

Final Projects

Getting Over It represents this addiction

$$\text{fall} + \gamma \text{ anger} + \ldots + \gamma^n \text{ win game} > \text{study} + \gamma \text{ exercise} + \ldots$$

It is always interesting to consider our own reward functions

Humans are born with reward functions and cannot choose them

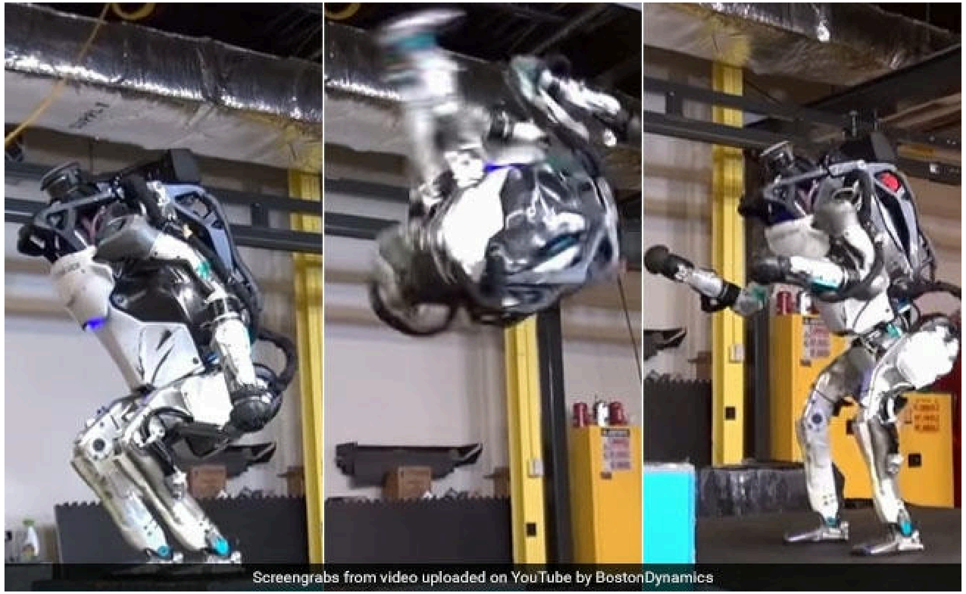
We can select good reward functions for artificial agents

LLMs are patient, intelligent, helpful because of the reward function

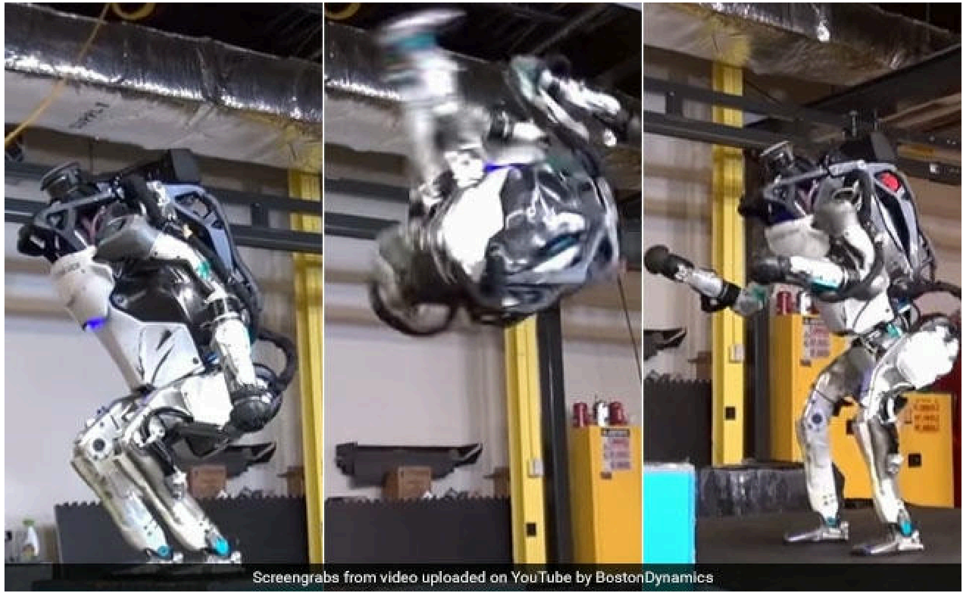
Finish DPG

Imitation Learning

Imitation Learning

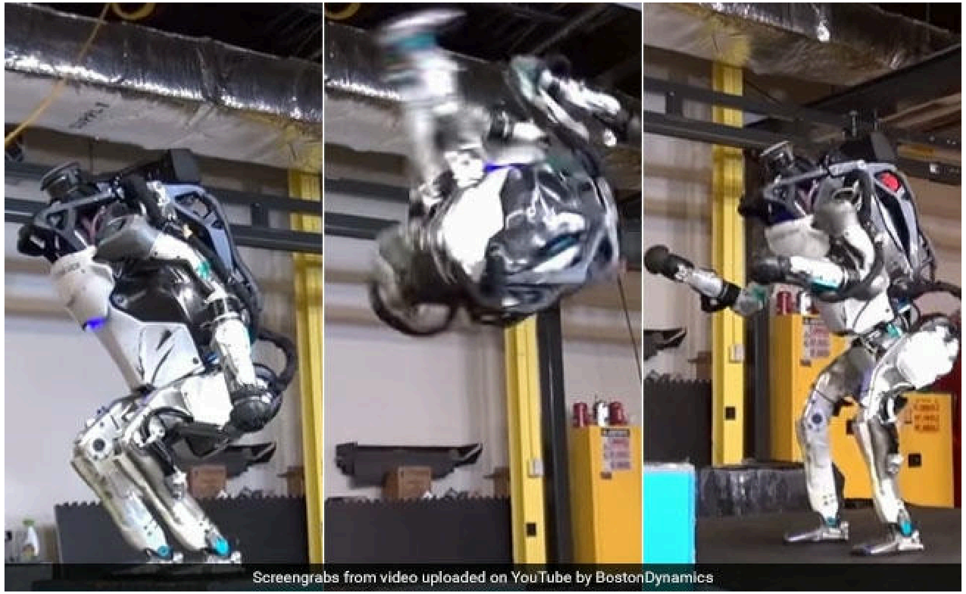


Imitation Learning



Example: You are a scientist at Boston Dynamics/Tencent Robotics/etc

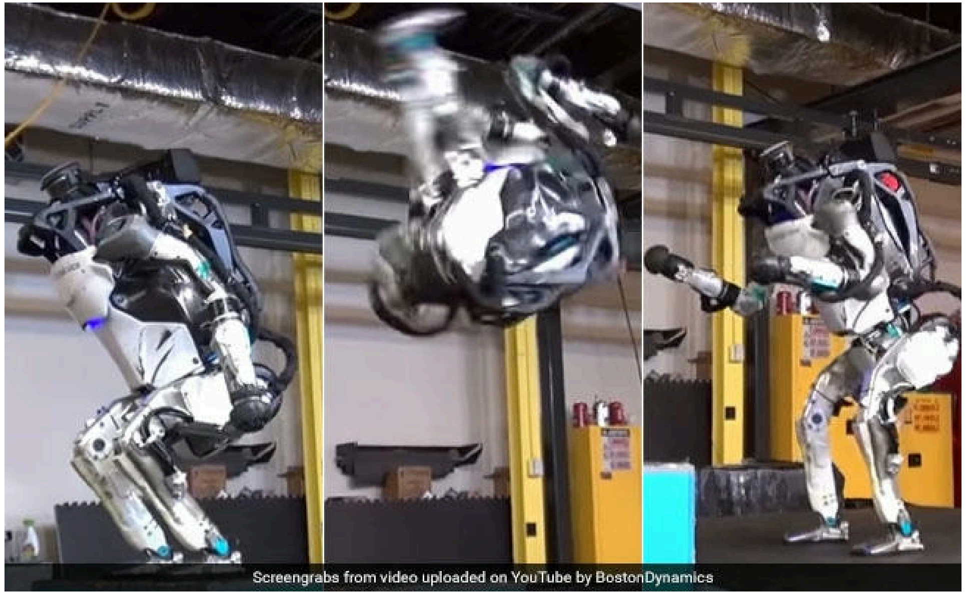
Imitation Learning



Example: You are a scientist at Boston Dynamics/Tencent Robotics/etc

You need to release demos to impress the public

Imitation Learning

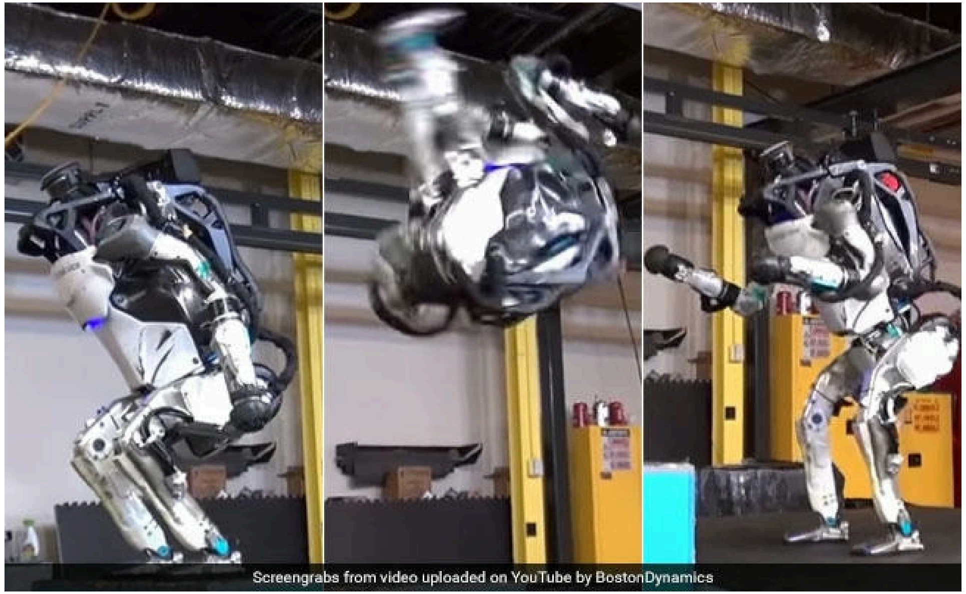


Example: You are a scientist at Boston Dynamics/Tencent Robotics/etc

You need to release demos to impress the public

You want to learn a backflip policy using RL

Imitation Learning



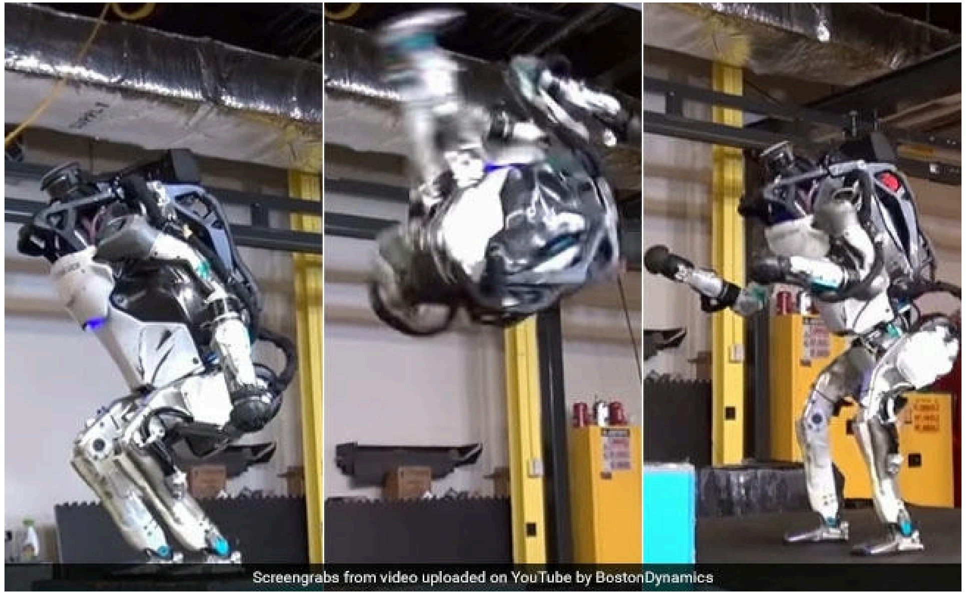
Example: You are a scientist at Boston Dynamics/Tencent Robotics/etc

You need to release demos to impress the public

You want to learn a backflip policy using RL

Question: What reward function should you use?

Imitation Learning



Example: You are a scientist at Boston Dynamics/Tencent Robotics/etc

You need to release demos to impress the public

You want to learn a backflip policy using RL

Question: What reward function should you use?

$$\mathcal{R}(s) =$$

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

We want our robot to:

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

We want our robot to:

- Be friendly

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

We want our robot to:

- Be friendly
- Be polite

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

We want our robot to:
- Be friendly
- Be polite
- Not scare humans

Imitation Learning

Policy will maximize the return, but not exhibit desired behavior

Writing reward functions can be difficult

Rewards are particularly difficult when interacting with humans

We want our robot to:

- Be friendly
- Be polite
- Not scare humans

It is very hard to write a reward function for these behaviors

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

How would a human do a backflip?

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

How would a human do a backflip?

$$\pi(a_0 \mid s_0, \theta_{\text{human}}), \pi(a_1 \mid s_1, \theta_{\text{human}}), \pi(a_2 \mid s_2, \theta_{\text{human}})...$$

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

How would a human do a backflip?

$$\pi(\underbrace{a_0}_{\text{Jump}} \mid s_0, \theta_{\text{human}}), \pi(\underbrace{a_1}_{\text{Tuck legs}} \mid s_1, \theta_{\text{human}}), \pi(\underbrace{a_2}_{\text{Backward rotation}} \mid s_2, \theta_{\text{human}})...$$

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

How would a human do a backflip?

Jump

Backward rotation

$$\pi(\boxed{a_0} \mid s_0, \theta_{\text{human}}), \pi(\boxed{a_1} \mid s_1, \theta_{\text{human}}), \pi(\boxed{a_2} \mid s_2, \theta_{\text{human}}) \ldots$$

Tuck legs

In **imitation learning** we learn to imitate an expert policy

Imitation Learning

It is often easier to demonstrate good behavior than create rewards

How would a human do a backflip?

$$\pi(\underbrace{a_0}_{\text{Jump}} \mid s_0, \theta_{\text{human}}), \pi(\underbrace{a_1}_{\text{Tuck legs}} \mid s_1, \theta_{\text{human}}), \pi(\underbrace{a_2}_{\text{Backward rotation}} \mid s_2, \theta_{\text{human}})...$$

In **imitation learning** we learn to imitate an expert policy

I want to introduce a formalism to model the problem

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function
- Dataset $\boldsymbol{X}$ from following an expert policy $\pi(a \mid s; \theta_\beta)$

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function
- Dataset $\boldsymbol{X}$ from following an expert policy $\pi\left(a \mid s; \theta_\beta\right)$

$$(S, A, \mathcal{R}, \mathrm{Tr}, \gamma, \boldsymbol{X})$$

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function
- Dataset $\boldsymbol{X}$ from following an expert policy $\pi\left(a \mid s; \theta_\beta\right)$

$$(S, A, \mathcal{R}, \mathrm{Tr}, \gamma, \boldsymbol{X})$$

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function
- Dataset $\boldsymbol{X}$ from following an expert policy $\pi\left(a \mid s; \theta_\beta\right)$

$$(S, A, \mathcal{R}, \mathrm{Tr}, \gamma, \boldsymbol{X})$$

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

$$\mathcal{R}(s_{t+1}) = ?$$

Imitation Learning

Definition: A Hidden-Reward MDP (HR-MDP) consists of:

- MDP with a **hidden** reward function
- Dataset $\boldsymbol{X}$ from following an expert policy $\pi(a \mid s; \theta_\beta)$

$$(S, A, \mathcal{R}, \mathrm{Tr}, \gamma, \boldsymbol{X})$$

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t, a_t)$$

$$\mathcal{R}(s_{t+1}) = ?$$

$$\boldsymbol{X} = [\boldsymbol{\tau}_1 \quad \boldsymbol{\tau}_2 \quad \dots] = \begin{bmatrix} \begin{bmatrix} (s_0, a_0) \\ (s_1, a_1) \\ \vdots \end{bmatrix} & \begin{bmatrix} (s_0, a_0) \\ (s_1, a_1) \\ \vdots \end{bmatrix} & \dots \end{bmatrix}$$

Imitation Learning

https://www.youtube.com/watch?v=4N4czAm61Fc

Behavioral Cloning

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

$$\arg\min_{\theta_\pi} \left(\pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\pi)\right)^2$$

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

$$\arg\min_{\theta_\pi} \left(\pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\pi)\right)^2$$

Question: Does this work?

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

$$\arg\min_{\theta_\pi} \left(\pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\pi)\right)^2$$

Question: Does this work?

Answer: No, $\pi(a \mid s)$ is a distribution not a scalar

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

$$\arg\min_{\theta_\pi} \left(\pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\pi)\right)^2$$

Question: Does this work?

Answer: No, $\pi(a \mid s)$ is a distribution not a scalar

Question: How can we measure the difference of two distributions?

Behavioral Cloning

Expert policy $\pi(a \mid s; \theta_\beta)$

Question: What is our objective?

$$\pi(a \mid s; \theta_\pi) = \pi(a \mid s; \theta_\beta)$$

$$\arg\min_{\theta_\pi} \left(\pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\pi)\right)^2$$

Question: Does this work?

Answer: No, $\pi(a \mid s)$ is a distribution not a scalar

Question: How can we measure the difference of two distributions?

Answer: KL divergence measures difference in distributions

Behavioral Cloning

$$\mathrm{KL}(\Pr(X), \Pr(Y)) = \sum_{\omega \in \Omega_X} \Pr(X = \omega) \log \frac{\Pr(X = \omega)}{\Pr(Y = \omega)}$$

Behavioral Cloning

$$\mathrm{KL}(\Pr(X), \Pr(Y)) = \sum_{\omega \in \Omega_X} \Pr(X = \omega) \log \frac{\Pr(X = \omega)}{\Pr(Y = \omega)}$$

Minimize difference between $\underbrace{\pi\left(a \mid s; \theta_\beta\right)}_{\Pr(X)}$ and $\underbrace{\pi\left(a \mid s; \theta_\pi\right)}_{\Pr(Y)}$

Behavioral Cloning

$$\mathrm{KL}(\Pr(X), \Pr(Y)) = \sum_{\omega \in \Omega_X} \Pr(X = \omega) \log \frac{\Pr(X = \omega)}{\Pr(Y = \omega)}$$

Minimize difference between $\underbrace{\pi\left(a \mid s; \theta_\beta\right)}_{\Pr(X)}$ and $\underbrace{\pi(a \mid s; \theta_\pi)}_{\Pr(Y)}$

$$\arg\min_{\theta_\pi} \mathrm{KL}\left(\pi\left(a \mid s; \theta_\beta\right), \pi(a \mid s; \theta_\pi)\right)$$

Behavioral Cloning

$$\mathrm{KL}(\Pr(X), \Pr(Y)) = \sum_{\omega \in \Omega_X} \Pr(X = \omega) \log \frac{\Pr(X = \omega)}{\Pr(Y = \omega)}$$

Minimize difference between $\underbrace{\pi\left(a \mid s; \theta_\beta\right)}_{\Pr(X)}$ and $\underbrace{\pi(a \mid s; \theta_\pi)}_{\Pr(Y)}$

$$\arg\min_{\theta_\pi} \; \mathrm{KL}\left(\pi\left(a \mid s; \theta_\beta\right), \pi(a \mid s; \theta_\pi)\right)$$

Plug policies into KL equation

Behavioral Cloning

$$\mathrm{KL}(\Pr(X), \Pr(Y)) = \sum_{\omega \in \Omega_X} \Pr(X = \omega) \log \frac{\Pr(X = \omega)}{\Pr(Y = \omega)}$$

Minimize difference between $\underbrace{\pi(a \mid s; \theta_\beta)}_{\Pr(X)}$ and $\underbrace{\pi(a \mid s; \theta_\pi)}_{\Pr(Y)}$

$$\arg\min_{\theta_\pi} \ \mathrm{KL}(\pi(a \mid s; \theta_\beta), \pi(a \mid s; \theta_\pi))$$

Plug policies into KL equation

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Log of divisors is difference of logs

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Log of divisors is difference of logs

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \left[\log(\pi(a \mid s; \theta_\beta)) - \log(\pi(a \mid s; \theta_\pi))\right]$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Log of divisors is difference of logs

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \left[\log(\pi(a \mid s; \theta_\beta)) - \log(\pi(a \mid s; \theta_\pi))\right]$$

Distribute behavior policy into difference

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \frac{\pi(a \mid s; \theta_\beta)}{\pi(a \mid s; \theta_\pi)}$$

Log of divisors is difference of logs

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \left[\log(\pi(a \mid s; \theta_\beta)) - \log(\pi(a \mid s; \theta_\pi))\right]$$

Distribute behavior policy into difference

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Can ignore the first term for optimization purposes

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Can ignore the first term for optimization purposes

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Can ignore the first term for optimization purposes

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

This is the **cross-entropy** objective

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Can ignore the first term for optimization purposes

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

This is the **cross-entropy** objective

Question: Have we seen this objective in deep learning?

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\beta) - \pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

First term is constant with respect to θ_π (only depends on θ_β)

Can ignore the first term for optimization purposes

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

This is the **cross-entropy** objective

Question: Have we seen this objective in deep learning?

Answer: Loss for classification tasks

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: Where does s come from?

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: Where does s come from? **Answer:** Dataset $\boldsymbol{X}$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: Where does s come from? **Answer:** Dataset $\boldsymbol{X}$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: Where does s come from? **Answer:** Dataset $\boldsymbol{X}$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

Definition: Behavioral cloning learns the parameters θ_π to match an expert policy θ_β that collected the dataset $\boldsymbol{X}$

Behavioral Cloning

Definition: Behavioral cloning learns the parameters θ_π to match an expert policy θ_β that collected the dataset $\boldsymbol{X}$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

Definition: Behavioral cloning learns the parameters θ_π to match an expert policy θ_β that collected the dataset $\boldsymbol{X}$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral cloning is a simple supervised learning algorithm!

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

If A is discrete, we can sum over possible actions $a \in A$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

If A is discrete, we can sum over possible actions $a \in A$

- Our expert policy will often be one-hot

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

If A is discrete, we can sum over possible actions $a \in A$

- Our expert policy will often be one-hot
- $\pi(a = a_* \mid s; \theta_\beta) = 1$ (action the expert took)

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

If A is discrete, we can sum over possible actions $a \in A$

- Our expert policy will often be one-hot
- $\pi(a = a_* \mid s; \theta_\beta) = 1$ (action the expert took)
- $\pi(a \neq a_* \mid s; \theta_\beta) = 0$ (all other actions)

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Question: How do we evaluate this?

If A is discrete, we can sum over possible actions $a \in A$

- Our expert policy will often be one-hot
- $\pi(a = a_* \mid s; \theta_\beta) = 1$ (action the expert took)
- $\pi(a \neq a_* \mid s; \theta_\beta) = 0$ (all other actions)

What if A is continuous (infinite)?

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

If A is continuous, we have an infinite sum

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

If A is continuous, we have an infinite sum

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

If A is continuous, we have an infinite sum

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Not all integrals are solvable

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \sum_{a \in A} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

If A is continuous, we have an infinite sum

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Not all integrals are solvable

Need to be careful how we model π, to make sure we can solve this

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

- Use Dirac delta expert policy $\pi(a \mid s; \theta_\beta) = \delta(a - a_*)$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi\left(a \mid s; \theta_\beta\right) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

- Use Dirac delta expert policy $\pi\left(a \mid s; \theta_\beta\right) = \delta(a - a_*)$
- Use Gaussian learned policy $\pi(a \mid s; \theta_\pi) = \mathrm{Normal}(\mu, \sigma)$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi\left(a \mid s; \theta_\beta\right) \log \pi\left(a \mid s; \theta_\pi\right) \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

- Use Dirac delta expert policy $\pi\left(a \mid s; \theta_\beta\right) = \delta(a - a_*)$
- Use Gaussian learned policy $\pi(a \mid s; \theta_\pi) = \mathrm{Normal}(\mu, \sigma)$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

- Use Dirac delta expert policy $\pi(a \mid s; \theta_\beta) = \delta(a - a_*)$
- Use Gaussian learned policy $\pi(a \mid s; \theta_\pi) = \mathrm{Normal}(\mu, \sigma)$

Plug our policies into the cross entropy objective

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

To ensure we have an analytical solution to this integral:

- Use Dirac delta expert policy $\pi(a \mid s; \theta_\beta) = \delta(a - a_*)$
- Use Gaussian learned policy $\pi(a \mid s; \theta_\pi) = \mathrm{Normal}(\mu, \sigma)$

Plug our policies into the cross entropy objective

Expert action taken

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\underbrace{\delta(a - a_*)}_{\text{Expert policy}} \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\underbrace{\delta(a - a_*)}_{\text{Expert policy}} \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Question: What happens to $\int_A \delta(a - a_*) f(a) \, \mathrm{d}a$?

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A -\underbrace{\delta(a - a_*)}_{\text{Expert policy}} \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Question: What happens to $\int_A \delta(a - a_*) f(a) \, \mathrm{d}a$?

Answer: Becomes $f(a = a_*)$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \int_A - \underbrace{\delta(a - a_*)}_{\text{Expert policy}} \log \pi(a \mid s; \theta_\pi) \, \mathrm{d}a$$

Question: What happens to $\int_A \delta(a - a_*) f(a) \, \mathrm{d}a$?

Answer: Becomes $f(a = a_*)$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \cancel{\int_A - \delta(a - a_*)} \log \pi(a \mid s; \theta_\pi) \cancel{\mathrm{d}a}$$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \pi(a = a_* \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log \pi(a = a_* \mid s; \theta_\pi)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log \pi(a = a_* \mid s; \theta_\pi)$$

Now, plug in Gaussian distribution for learned policy

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \pi(a = a_* \mid s; \theta_\pi)$$

Now, plug in Gaussian distribution for learned policy

$$\pi(a = a_* \mid s; \theta_\pi) = \mathrm{Normal}(a = a_* \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \pi(a = a_* \mid s; \theta_\pi)$$

Now, plug in Gaussian distribution for learned policy

$$\pi(a = a_* \mid s; \theta_\pi) = \mathrm{Normal}(a = a_* \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right)$$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right) \right]$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right) \right]$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right]$$

Log of products is sum of logs

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2} \right) \right]$$

Log of products is sum of logs

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2} \right) \right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right]$$

Log of products is sum of logs

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Log of divisors is difference of logs

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right) \right]$$

Log of products is sum of logs

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \log\left(\exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right) \right)$$

Log of divisors is difference of logs

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log(1) + \log\left(\sqrt{2\pi\sigma^2} \right) - \log\left(\exp\left(- \frac{(a_* - \mu)^2}{2\sigma^2} \right) \right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log(1) + \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log(1) + \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

$$\log(1) = 0$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} -\log(1) + \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

$\log(1) = 0$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log(1) + \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

$\log(1) = 0$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Rewrite $\sqrt{...}$ as power

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} - \log(1) + \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

$\log(1) = 0$

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\sqrt{2\pi\sigma^2}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Rewrite $\sqrt{...}$ as power

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\left(2\pi\sigma^2\right)^{\frac{1}{2}}\right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2}\right)\right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\left(2\pi\sigma^2\right)^{\frac{1}{2}}\right) - \log\left(\exp\left(-\frac{\left(a_* - \mu\right)^2}{2\sigma^2}\right)\right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\left(2\pi\sigma^2\right)^{\frac{1}{2}} \right) - \log\left(\exp\left(-\frac{\left(a_* - \mu\right)^2}{2\sigma^2} \right) \right)$$

Log of power is product of power and log

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left((2\pi\sigma^2)^{\frac{1}{2}} \right) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2} \right) \right)$$

Log of power is product of power and log

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) - \log\left(\exp\left(-\frac{(a_* - \mu)^2}{2\sigma^2} \right) \right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\left(2 \pi \sigma^2\right)^{\frac{1}{2}} \right) - \log\left(\exp\left(- \frac{\left(a_* - \mu\right)^2}{2 \sigma^2} \right) \right)$$

Log of power is product of power and log

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log\left(2 \pi \sigma^2\right) - \log\left(\exp\left(- \frac{\left(a_* - \mu\right)^2}{2 \sigma^2} \right) \right)$$

Now simplify the second term, log and exp cancel

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \log\left(\left(2\pi\sigma^2\right)^{\frac{1}{2}} \right) - \log\left(\exp\left(- \frac{\left(a_* - \mu\right)^2}{2\sigma^2} \right) \right)$$

Log of power is product of power and log

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log\left(2\pi\sigma^2\right) - \log\left(\exp\left(- \frac{\left(a_* - \mu\right)^2}{2\sigma^2} \right) \right)$$

Now simplify the second term, log and exp cancel

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log\left(2\pi\sigma^2\right) + \frac{\left(a_* - \mu\right)^2}{2\sigma^2}$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Since this is an optimization problem, we can drop the constant 2π

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Since this is an optimization problem, we can drop the constant 2π

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log \sigma^2 + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Since this is an optimization problem, we can drop the constant 2π

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log \sigma^2 + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Clean up a bit by factoring

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Since this is an optimization problem, we can drop the constant 2π

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log \sigma^2 + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Clean up a bit by factoring

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Behavioral Cloning

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log(2\pi\sigma^2) + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Since this is an optimization problem, we can drop the constant 2π

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \log \sigma^2 + \frac{(a_* - \mu)^2}{2\sigma^2}$$

Clean up a bit by factoring

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Behavioral Cloning

For a Gaussian policy, we minimize the following objective

Behavioral Cloning

For a Gaussian policy, we minimize the following objective

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Behavioral Cloning

For a Gaussian policy, we minimize the following objective

$$\underset{\theta_\pi}{\arg\min} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Note: We derive the loss function for a Gaussian policy gradient loss the exact same way

Behavioral Cloning

Definition: Behavioral cloning uses **supervised learning** for decision making, minimizing the cross-entropy between expert θ_β and our θ_π

$$\arg\min_{\theta_\pi} \sum_{s,a \in \boldsymbol{X}} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

For continuous actions, we pick special distributions so the cross entropy loss is tractable (Dirac delta θ_β, Gaussian θ_π)

Behavioral Cloning

Definition: Behavioral cloning uses **supervised learning** for decision making, minimizing the cross-entropy between expert θ_β and our θ_π

$$\arg\min_{\theta_\pi} \sum_{s,a \in \boldsymbol{X}} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

For continuous actions, we pick special distributions so the cross entropy loss is tractable (Dirac delta θ_β, Gaussian θ_π)

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Behavioral Cloning

Definition: Behavioral cloning uses **supervised learning** for decision making, minimizing the cross-entropy between expert θ_β and our θ_π

$$\arg\min_{\theta_\pi} \sum_{s, a \in \boldsymbol{X}} -\pi(a \mid s; \theta_\beta) \log \pi(a \mid s; \theta_\pi)$$

For continuous actions, we pick special distributions so the cross entropy loss is tractable (Dirac delta θ_β, Gaussian θ_π)

Expert action

$$\arg\min_{\theta_\pi} \sum_{s \in \boldsymbol{X}} \frac{1}{2} \left(\log \sigma^2 + \frac{(a_* - \mu)^2}{\sigma^2} \right)$$

Policy outputs

Coding

Coding

Similar to policy gradient, just with different loss function

Coding

Similar to policy gradient, just with different loss function

The policies and methods change depending on action space

Coding

Similar to policy gradient, just with different loss function

The policies and methods change depending on action space

Discrete actions use a categorical distribution

Coding

Similar to policy gradient, just with different loss function

The policies and methods change depending on action space

Discrete actions use a categorical distribution

Continuous actions usually use normal distribution

Coding

Similar to policy gradient, just with different loss function

The policies and methods change depending on action space

Discrete actions use a categorical distribution

Continuous actions usually use normal distribution

Start with discrete actions, then do continous

Coding

Implement a model for a categorical policy

Coding

Implement a model for a categorical policy

```
model = Sequential([
    Linear(state_size, hidden_size),
    Lambda(leaky_relu),
    Linear(hidden_size, hidden_size),
    Lambda(leaky_relu),
    # Output logits (real numbers)
    Linear(hidden_size, action_size),
])
```

Coding

Implement a model for a categorical policy

```
model = Sequential([
    Linear(state_size, hidden_size),
    Lambda(leaky_relu),
    Linear(hidden_size, hidden_size),
    Lambda(leaky_relu),
    # Output logits (real numbers)
    Linear(hidden_size, action_size),
])
```

Identical to policy gradient

Coding

Next, implement discrete loss function

Coding

Next, implement discrete loss function

```
def bc_loss(model, states, actions):
    # Often, we can't know the expert action distribution
    # We only have the taken expert action
    # Taken action has p=1, all other actions p=0
    # Represent as a one-hot vector
    expert_probs = actions
    log_policy_probs = log_softmax(vmap(model)(states))
    # Log loss, can reduce over batch using mean or sum
    bce_loss = -sum(
        expert_probs * log_policy_probs, axis=1).mean()
    return bce_loss
```

Coding

Finally, to run our policy we sample actions from our policy

Coding

Finally, to run our policy we sample actions from our policy

```
def sample_action(model, state, key):
    z = model(state)
    # BE VERY CAREFUL, always read documentation
    # Sometimes takes UNNORMALIZED logits, sometimes probs
    action_probs = softmax(model, state)
    a = categorical(key, action_probs)
    a = categorical(key, z) # Does not even use pi
    return a
```

Coding

Finally, to run our policy we sample actions from our policy

```
def sample_action(model, state, key):
    z = model(state)
    # BE VERY CAREFUL, always read documentation
    # Sometimes takes UNNORMALIZED logits, sometimes probs
    action_probs = softmax(model, state)
    a = categorical(key, action_probs)
    a = categorical(key, z) # Does not even use pi
    return a
```

Identical to policy gradient

Coding

Now, consider Gaussian policy (continuous actions)

Coding

Now, consider Gaussian policy (continuous actions)

```
model = Sequential([
    Linear(state_size, hidden_size),
    Lambda(leaky_relu),
    Linear(hidden_size, hidden_size),
    Lambda(leaky_relu),
    # Output mu and log_sigma
    Linear(hidden_size, 2 * action_size),
    Lambda(x: split(x, 2))
])
```

Coding

Now, consider Gaussian policy (continuous actions)

```
model = Sequential([
    Linear(state_size, hidden_size),
    Lambda(leaky_relu),
    Linear(hidden_size, hidden_size),
    Lambda(leaky_relu),
    # Output mu and log_sigma
    Linear(hidden_size, 2 * action_size),
    Lambda(x: split(x, 2))
])
```

Same as policy gradient

Coding

Implement continuous loss function

Coding

Implement continuous loss function

Use simplified cross entropy (Dirac-Gaussian)

Coding

Implement continuous loss function

Use simplified cross entropy (Dirac-Gaussian)

```
def bc_loss(model, states, actions):
    expert_probs = actions # Dirac delta
    mu, log_std = vmap(model)(states)
    # Gaussian CE, also called Gaus. Neg. Log Likelihood
    gnll_loss = log_std + 0.5 * (
        (mu - action)**2 / exp(log_std)**2
    )
    return gnll_loss
```

Coding

Next, we need to sample actions from our policy network

```
def sample_action(model, state, key):
    mu, log_sigma = model(state)
    # Reparameterization trick
    noise = random.normal(key, (action_size,))
    a = mu + exp(log_sigma) * noise
    return a
```

Coding

The training loop is much simpler than RL

Coding

The training loop is much simpler than RL

```
model = Sequential(...)
opt_state = ...
# Just supervised learning
for batch in dataset:
    states, actions = batch
    J = grad(bc_loss)(model, states, actions)
    update = optim.update(J, opt_state)
    model = apply_updates(update, model)
```

Coding

The training loop is much simpler than RL

```
model = Sequential(...)
opt_state = ...
# Just supervised learning
for batch in dataset:
    states, actions = batch
    J = grad(bc_loss)(model, states, actions)
    update = optim.update(J, opt_state)
    model = apply_updates(update, model)
```

Then deploy the policy

Coding

The training loop is much simpler than RL

```
model = Sequential(...)
opt_state = ...
# Just supervised learning
for batch in dataset:
    states, actions = batch
    J = grad(bc_loss)(model, states, actions)
    update = optim.update(J, opt_state)
    model = apply_updates(update, model)
```

Then deploy the policy

```
a_t = sample_action(model, s_t, key)
```

Applications

Applications

Behavioral cloning seems very powerful!

Applications

Behavioral cloning seems very powerful!

- No need for reward function

Applications

Behavioral cloning seems very powerful!

- No need for reward function
- No need for exploration, learn from fixed dataset

Applications

Behavioral cloning seems very powerful!

- No need for reward function
- No need for exploration, learn from fixed dataset
- Simpler to implement than RL

Applications

Behavioral cloning seems very powerful!

- No need for reward function
- No need for exploration, learn from fixed dataset
- Simpler to implement than RL
- More stable to train than RL

Applications

Behavioral cloning seems very powerful!

- No need for reward function
- No need for exploration, learn from fixed dataset
- Simpler to implement than RL
- More stable to train than RL

Sounds very promising, why do we care about RL?

Applications

Behavioral cloning seems very powerful!

- No need for reward function
- No need for exploration, learn from fixed dataset
- Simpler to implement than RL
- More stable to train than RL

Sounds very promising, why do we care about RL?

Question: What are some disadvantages of BC?

Applications

Limitation: Imperfect expert

Applications

Limitation: Imperfect expert

Dataset is following an “expert” θ_β

Applications

Limitation: Imperfect expert

Dataset is following an “expert” θ_β

Humans are not reliable experts in many cases

Applications

Limitation: Imperfect expert

Dataset is following an "expert" θ_β

Humans are not reliable experts in many cases

You will learn a policy that drives like it is texting

Applications

Even where all the data is from a reliable “expert”, we have problems

Applications

Even where all the data is from a reliable “expert”, we have problems

Consider a human surgeon

Applications

Even where all the data is from a reliable "expert", we have problems

Consider a human surgeon

Human surgeons are not as fast or precise as machines

Applications

Even where all the data is from a reliable “expert”, we have problems

Consider a human surgeon

Human surgeons are not as fast or precise as machines

Even the best machine will be no better than a human surgeon

Applications

Even where all the data is from a reliable "expert", we have problems

Consider a human surgeon

Human surgeons are not as fast or precise as machines

Even the best machine will be no better than a human surgeon

$$\pi(a \mid s; \theta_\pi) \approx \underbrace{\pi\left(a \mid s; \theta_\beta\right)}_{\text{Human policy}}$$

Applications

Even where all the data is from a reliable "expert", we have problems

Consider a human surgeon

Human surgeons are not as fast or precise as machines

Even the best machine will be no better than a human surgeon

$$\pi(a \mid s; \theta_\pi) \approx \underbrace{\pi\left(a \mid s; \theta_\beta\right)}_{\text{Human policy}}$$

Question: Any other issues?

Applications

Limitation: Humans limited to small regions of state space

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

There will be very few crashes in a self driving dataset

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

There will be very few crashes in a self driving dataset

When the policy is about to crash, the state will be out of distribution

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

There will be very few crashes in a self driving dataset

When the policy is about to crash, the state will be out of distribution

The policy will be unable to avoid crashing

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

There will be very few crashes in a self driving dataset

When the policy is about to crash, the state will be out of distribution

The policy will be unable to avoid crashing

In practice, BC policies generalizes much worse than RL policies

Applications

Limitation: Humans limited to small regions of state space

Humans are very bad at exploring

There will be very few crashes in a self driving dataset

When the policy is about to crash, the state will be out of distribution

The policy will be unable to avoid crashing

In practice, BC policies generalizes much worse than RL policies

Small errors in the learned policy eventually drive the policy to out of distribution states

Applications

One solution to out of distribution error is to collect more data

Applications

One solution to out of distribution error is to collect more data

DAgger: The agent asks an expert for help when it visit an out of distribution state

Applications

One solution to out of distribution error is to collect more data

DAgger: The agent asks an expert for help when it visit an out of distribution state

- This requires an expert to always be paying attention

Applications

One solution to out of distribution error is to collect more data

DAgger: The agent asks an expert for help when it visit an out of distribution state

- This requires an expert to always be paying attention

Inverse RL: We learn the reward function that the expert is maximizing

Applications

One solution to out of distribution error is to collect more data

DAgger: The agent asks an expert for help when it visit an out of distribution state

- This requires an expert to always be paying attention

Inverse RL: We learn the reward function that the expert is maximizing

$$\arg\max_{\theta_R} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\beta\right]$$

$$= \arg\max_{\theta_R} \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \underbrace{\mathcal{R}(s_{n+1}, \theta_R)}_{\text{Learnable}} \cdot \Pr(s_{n+1} \mid s_0; \theta_\beta)$$

Applications

One solution to out of distribution error is to collect more data

DAgger: The agent asks an expert for help when it visit an out of distribution state

- This requires an expert to always be paying attention

Inverse RL: We learn the reward function that the expert is maximizing

$$\arg\max_{\theta_R} \mathbb{E}\left[\mathcal{G}(\boldsymbol{\tau}) \mid s_0; \theta_\beta\right]$$

$$= \arg\max_{\theta_R} \sum_{n=0}^{\infty} \gamma^n \sum_{s_{n+1} \in S} \underbrace{\mathcal{R}(s_{n+1}, \theta_R)}_{\text{Learnable}} \cdot \Pr(s_{n+1} \mid s_0; \theta_\beta)$$

Then, we learn a policy θ_π using RL. This generalizes better than BC