Imitation Learning

CS 224R
Course reminders

- Start forming final project groups (survey due Mon April 17)
- Homework 1 out today, due Weds April 19
- Fill out AWS form with account ID by this Friday April 7

News

- Thursday PyTorch tutorial (4:30 pm) moved to Skilling Auditorium
- Up to 2% extra credit for providing TA-endorsed answers on Ed
The plan for today

Imitation Learning
1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
A formalization of behavior

Observe state $s_t$

Take action $a_t$ (e.g. by sampling from policy $\pi_\theta(\cdot | s_t)$)

Observe next state $s_{t+1}$ sampled from unknown world dynamics $p(\cdot | s_t, a_t)$

Result: a trajectory $s_1, a_1, \ldots, s_T$. also called a policy roll-out
The basics of imitation learning

**Key idea:** Train policy using supervised learning

**Data:** Given trajectories collected by an expert

“demonstrations” \( \mathcal{D} := \{(s_1, a_1, \ldots, s_T)\} \)

**Training:** Train policy to mimic expert: \( \min_{\theta} - \mathbb{E}_{(s,a) \sim \mathcal{D}}[\log \pi_\theta(a | s)] \)

i.e. minimize cross-entropy loss or \( \ell_2 \) loss between predicted & expert actions.
The plan for today

Imitation Learning

1. *Where does the data come from?*
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
How to collect demonstrations?

In some domains: People already collect demonstrations that can be recorded e.g. driving cars, writing text messages

What about robotics?

- Kinesthetic teaching
  - + easy interface
  - - human visible in scene

- Remote controllers
  - ~ interface ease varies

- Puppeteering
  - + easy interface
  - - requires double hardware

In other domains: It may not be viable to collect demos! (e.g. quadruped robot)
Can we directly use videos of people, animals?

**Embodiment gap:**
- difference in appearance
- difference in physical capabilities, degrees of freedom

Hard to directly imitate human & animal data, but can guide exploration.

The plan for today

Imitation Learning
1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
What can go wrong in imitation learning?
What can go wrong in imitation learning?

1. Compounding errors

Supervised learning

Inputs independent of predicted labels $\hat{y}$

Supervised learning of behavior

Predicted actions affect next state.
Errors can lead to drift away from the data distribution!
Errors can then compound!

$p_{expert}(s) \neq p_\pi(s)$

states visited by expert
states visited by learned policy $\pi$

“covariate shift”
What can go wrong in imitation learning?

1. Compounding errors

Supervised learning of behavior

Predicted actions affect next state.

Errors can lead to drift away from the data distribution!

Errors can then compound!

\[ p_{\text{expert}}(s) \neq p_{\pi}(s) \]

Solutions?

1. Collect A LOT of demo data & hope for the best.
2. Collect corrective behavior data
What can go wrong in imitation learning?

2. Multimodal demonstration data

The data takes two different actions here!

If we use $\ell_2$ loss, what action will the agent take?

When does this happen in practice? All time time! Esp. when data collected by multiple people.

Solution? Use expressive distribution class to fit $p(a \mid s)$.
- capture all modes of the data distribution
- e.g. Gaussian mixture, Categorical, VAEs, diffusion models
What can go wrong in imitation learning?

3. Mismatch in observability between expert & agent

Example demos scraped from conversations:

Problem: Expert has more information than is observed by the agent.

S
Hi, how are you?

a
Great, how was the basketball game last weekend?

... Impossible to accurately imitate.

S
Hey, how are you?

a
I’m good. Looking forward to getting lunch tomorrow!

... Solutions:

- Give as much contextual information to the agent as possible.
- Collect demos in a way that gives expert same information as agent.
What can go wrong in imitation learning?

1. Compounding errors
2. Multimodal demonstration data
3. Mismatch in observability between expert & agent
The plan for today

Imitation Learning
1. Where does the data come from?
2. What can go wrong?
3. **Learning from online interventions**
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
Addressing Compounding Errors with DAgger

Collect corrective behavior data

1. Roll-out learned policy \( \pi_\theta; s_1', \hat{a}_1, \ldots, s'_T \)
2. Query expert action at visited states \( a^* \sim \pi_{\text{expert}}(\cdot | s') \)
3. Aggregate corrections with existing data \( \mathcal{D} \leftarrow \mathcal{D} \cup \{ (s', a^*) \} \)
4. Update policy \( \min_{\theta} \mathcal{L}(\pi_\theta, \mathcal{D}) \)

"dataset aggregation" (DAgger)

+ data-efficient way to learn from an expert
- can be challenging to query expert when agent has control

Is there another way to collect corrective data?
Addressing Compounding Errors with DAgger

Collect corrective behavior data while *taking full control*

1. Start to roll-out learned policy $\pi_\theta$: $s_1', \hat{a}_1, \ldots, s_t'$
2. Expert intervenes at time $t$ when policy makes mistake
3. Expert provides (partial) demonstration $s_t', a^*_t, \ldots, s_T'$
4. Aggregate new demos with existing data $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_i', a^*_i)\}; i \geq t$
5. Update policy $\min_\theta \mathcal{L}(\pi_\theta, \mathcal{D})$

“human gated DAgger”

+ (much) easier interface for providing corrections
- can be hard to catch mistakes quickly in some application domains
The plan for today

Imitation Learning

1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
Case study: Can robots learn fine-grained manipulation skills from demonstrations?

Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

Tony Z. Zhao¹ Vikash Kumar² Sergey Levine² Chelsea Finn¹
¹ Stanford University ² UC Berkeley ³ Meta

**Goal:** Solve tasks where *precision* and *closed-loop feedback* are important, with objects that are *difficult to simulate*

→ by learning from real-world data.
Hardware Set-Up

- Total cost: <$20k
- Off-the-shelf arms + open-sourced parts & code
- Fun to play with. :)}
- **Total cost**: <$20k
- Off-the-shelf arms + open-sourced parts & code
- Off-the-shelf 6-DoF arms with 3D-printed fingers
- Map joint angles across robots during teleoperation
- 50 Hz control
- Record RGB images from 4 cameras
- No force feedback (beyond weight of “leader” robot)
Challenge 1: Supervised imitation learning struggles with compounding errors, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to multimodal data distribution.

<table>
<thead>
<tr>
<th></th>
<th>Slide Ziploc (real)</th>
<th>Slot Battery (real)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Grasp</td>
<td>Pinch</td>
</tr>
<tr>
<td>BC-ConvMLP</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BeT</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>RT-1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>VINN</td>
<td>28</td>
<td>0</td>
</tr>
</tbody>
</table>

Success rates of carefully-tuned prior IL methods.
Challenge 1: Supervised imitation learning struggles with compounding errors, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to multimodal data distribution.

Solutions for #1:
- Policy predicts chunks of ~60 actions open-loop (closed-loop at ~0.8 Hz, rather than making new decision every timestep)

Action Chunking

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t=4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Naive policy training achieves 0% success.
Challenge 1: Supervised imitation learning struggles with compounding errors, particularly at 50 Hz.

Challenge 2: Human demonstrations perform tasks in different ways, leading to multimodal data distribution.

Solutions for #1:
- Policy predicts chunks of ~60 actions open-loop (closed-loop at ~0.8 Hz, rather than making new decision every timestep)
- Weighted average over predicted actions for that timestep

Naive policy training achieves 0% success.
Challenge 1: Supervised imitation learning struggles with compounding errors, particularly at 50 Hz. Naive policy training achieves 0% success.

Challenge 2: Human demonstrations perform tasks in different ways, leading to multimodal data distribution.

Solutions for #1:
- Policy predicts chunks of ~60 actions open-loop (closed-loop at ~0.8 Hz, rather than making new decision every timestep)  
  \{ trade-off drift & open-loop \}
- Weighted average over predicted actions for that timestep
- Transformer-based policy architecture
- Actions correspond to target absolute joint positions (rather than relative joint positions)

Solution for #2:
- Use variational auto-encoder (VAE) to model multimodality
Imitation Learning System

Policy architecture

Action chunking with transformers (ACT)
Simulated Results

Grasp & transfer from image observations

How does ACT compare to prior methods?

<table>
<thead>
<tr>
<th>Method</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP policy</td>
<td>1%</td>
</tr>
<tr>
<td>Behavior Transformer (BeT)</td>
<td>27%</td>
</tr>
<tr>
<td>Visual Imitation Nearest Neighbors (VINN)</td>
<td>3%</td>
</tr>
<tr>
<td>RT-1</td>
<td>2%</td>
</tr>
<tr>
<td>ACT</td>
<td>86%</td>
</tr>
<tr>
<td>ACT, no action chunking</td>
<td>0%</td>
</tr>
</tbody>
</table>

Is action chunking important?
Real Robot Results

Collect 50 demonstrations, randomize object location along white line.

Success rate: 84%
Real Robot Results
Collect 50 demonstrations, randomize object location along white line.
Success rate: 96%
Real Robot Results
Collect 50 demonstrations, randomize object location along white line.
Success rate: 64%
Real Robot Results

Collect 50 demonstrations, randomize object location along white line.

Success rate: 92%
Simulated Ablations

Action chunking

Temporal aggregation

VAE

Scripted Data

Human Data
Simulated Ablations

### Action chunking

- Ours
- BC-ConvMLP
- VINN

### Temporal aggregation

- Ours
- BC-ConvMLP
- VINN

- **no TA**
- **with TA**

### VAE

- Scripted Data
- Human Data

- **With CVAE**
- **No CVAE**
Simulated Ablations

Action chunking

- fully-closed-loop
- fully-open-loop

Temporal aggregation

- Ours
- BC-ConvMLP
- VINN

- no TA
- with TA

VAE

- Scripted Data
- Human Data

With CVAE
- No CVAE
Recap

Imitation Learning
1. Where does the data come from?
2. What can go wrong?
3. Learning from online interventions
4. Case study in fine robotic manipulation

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them
Recap

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

**Data**: Given trajectories collected by an expert \( \mathcal{D} := \{(s_1, a_1, \ldots, s_T)\} \)

“demonstrations”

**Training**: Train policy to mimic expert: 

\[
\min_{\theta} - \mathbb{E}_{(s,a) \sim \mathcal{D}}[\log \pi_{\theta}(a | s)]
\]
Recap

Key learning goals:
- the basic mechanics of imitation learning & how to implement it
- the most common challenges & latest solutions for addressing them

Common Challenges:
1. Compounding errors
2. Multimodal demonstration data
3. Mismatch in observability

Some Solutions:
- more data, online interventions
- use more expressive distributions
- provide more context, or collect data with less context
Is Imitation Learning All You Need?

A simple & powerful framework for learning behavior!

**But:**
- Collecting expert demonstrations can be difficult or impossible in some scenarios
  - Learned behavior will never be better than expert
- Does not provide a framework for learning from experience, indirect feedback
  - Can agents learn autonomously, from their own mistakes?

**Next time:** Start of reinforcement learning algorithms
We’ll revisit imitation learning in week 4.
Course reminders

- **Start forming final project groups** (survey due Mon April 17)
- **Homework 1** out today, due Weds April 19
- Fill out **AWS form** with account ID by this Friday April 7

News

- Thursday PyTorch tutorial (4:30 pm) moved to **Skilling Auditorium**
- Up to **2% extra credit** for providing TA-endorsed answers on Ed