Behavioral Cloning and Interactive Imitation Learning

Instructor: Daniel Brown

[Some slides adapted from Sergey Levine (CS 285) and Alina Vereshchaka (CSE4/510)]
Course feedback is open

• Extra credit if class response rate is 70% or higher
  • Sliding scale if we reach 70%:
    • Extra credit points = response rate percentage / 10
Reinforcement Learning

Action

Observation

Reward
Reinforcement Learning

Action

Observation

Reward
Reward engineering is hard!
Reward engineering is hard!

Action

Observation

Reward
Reward engineering is hard!
Reinforcement learning is hard...even with a reward function!
Imitation Learning:
Learn a policy from examples of good behavior.

- Often showing is easier than telling.
- Alleviates problem of exploration.
Behavioral Cloning

What would the human do?

Policy $\pi$

1. Observation
2. Action
3. Disc actions, cont. actions
4. Classifier, regression

$D = \{(s_0, a_0), (s_1, a_1)\}$
Inverse Reinforcement Learning

Why? What is the human’s reward function?

We’ll talk about this on Thursday!
Imitation Learning via Behavioral Cloning

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

Supervised machine learning:

\[ f(x) = y \]

\[ \pi_\theta(a_t|o_t) \]
ALVINN: One of the first imitation learning systems

ALVINN: Autonomous Land Vehicle In a Neural Network
1989
ALVINN: One of the first imitation learning systems

ALVINN: Autonomous Land Vehicle In a Neural Network
1989
What could go wrong?
Distribution Shift

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

<table>
<thead>
<tr>
<th></th>
<th>Supervised Learning</th>
<th>Supervised Learning + Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>((x, y) \sim D)</td>
<td>(s \sim T(s, a, \pi^*(s)))</td>
</tr>
<tr>
<td>Test</td>
<td>((x, y) \sim D)</td>
<td>(s \sim T(s, a, \pi(s)))</td>
</tr>
</tbody>
</table>
But it still can work in practice...
How?

Bojarski et al. ‘16, NVIDIA
A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots

Alessandro Giusti¹, Jérôme Guzzi¹, Dan C. Cireşan¹, Fang-Lin He¹, Juan P. Rodríguez¹
Flavio Fontana², Matthias Faessler², Christian Forster²
Jürgen Schmidhuber¹, Gianni Di Caro¹, Davide Scaramuzza², Luca M. Gambardella¹
Can we make it work more often?

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

Can we make \( p_{\text{data}}(o_t) = p_{\pi_\theta}(o_t) \)?
human recovery policy
training trajectory
$\pi_\theta$ expected trajectory
DAgger

To *learned policy*

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

**DAgger: Dataset Aggregation**

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

1. train $\pi_\theta(a_t|o_t)$ from human data $\mathcal{D} = \{o_1, a_1, \ldots, o_N, a_N\}$
2. run $\pi_\theta(a_t|o_t)$ to get dataset $\mathcal{D}_\pi = \{o_1, \ldots, o_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $a_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$
DAgger has very nice theoretical guarantees.

Why might it be hard to implement in practice?

**DAgger: Dataset Aggregation**

goal: collect training data from $p_{\pi_\theta}(o_t)$ instead of $p_{\text{data}}(o_t)$

how? just run $\pi_\theta(a_t | o_t)$

but need labels $a_t$!

1. train $\pi_\theta(a_t | o_t)$ from human data $D = \{o_1, a_1, \ldots, o_N, a_N\}$
2. run $\pi_\theta(a_t | o_t)$ to get dataset $D_{\pi} = \{o_1, \ldots, o_M\}$
3. Ask human to label $D_{\pi}$ with actions $a_t$
4. Aggregate: $D \leftarrow D \cup D_{\pi}$

Ross et al. ‘11
1. train $\pi_\theta(a_t|o_t)$ from human data $D = \{o_1, a_1, \ldots, o_N, a_N\}$
2. run $\pi_\theta(a_t|o_t)$ to get dataset $D_\pi = \{o_1, \ldots, o_M\}$
3. Ask human to label $D_\pi$ with actions $a_t$
4. Aggregate: $D \leftarrow D \cup D_\pi$
DAgger has very nice theoretical guarantees.

Why might it be easy to implement in practice?

**DAgger: Dataset Aggregation**

goal: collect training data from $p_{\pi_{\theta}}(o_t)$ instead of $p_{\text{data}}(o_t)$

how? just run $\pi_{\theta}(a_t|o_t)$

but need labels $a_t$!

1. train $\pi_{\theta}(a_t|o_t)$ from human data $D = \{o_1, a_1, \ldots, o_N, a_N\}$
2. run $\pi_{\theta}(a_t|o_t)$ to get dataset $D_{\pi} = \{o_1, \ldots, o_M\}$
3. Ask human to label $D_{\pi}$ with actions $a_t$
4. Aggregate: $D \leftarrow D \cup D_{\pi}$
Learn from an Algorithmic Supervisor!

But we don’t always have access to an algorithmic supervisor...

Can we make DAgger more practical when dealing with real human labeling?
Interactive IL

$\pi_H(s)$

$\pi_{\text{meta}}(s)$

$\pi_R(s)$
Human-Gated Interactive IL

Robot-Gated Interactive IL

Minimizing Supervisor Burden

- $C = \text{Number of context switches}$
- $L = \text{Latency of context switching}$
- $I = \text{Expected number of supervisor actions per intervention}$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

Ideally, we want

$$\pi = \arg \min_{\pi' \in \Pi} L(\pi'_r)$$

$s.t. B(\pi') \leq \Gamma_b$
Minimizing Supervisor Burden

• $C = \text{number of context switches}$
• $L = \text{Latency of context switching}$
• $I = \text{expected number of supervisor actions per intervention}$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

In practice, we approximate this by focusing on limiting the number of interventions (number of context switches)

$$\pi = \arg\min_{\pi' \in \Pi} L(\pi'_r)$$

s.t. $B(\pi') \leq \Gamma_b$
SafeDAgger

Predicted action loss = predicted difference between human and robot action.

Trained using held-out set of data from human.

Hoque et al. 2021.
SafeDAgger

Supervisor Mode

Autonomous Mode

\( \beta_H \)

Predicted Action Loss

LazyDAgger

Supervisor Mode + Noise

Autonomous Mode

\( \beta_H \)

Predicted Action Loss

\( \beta_R \)

True Action Loss

Hysteresis

Hoque et al. 2021.
LazyDAgger

\[ s_t \rightarrow \text{Query Robot Action} \ a_t = \pi_R(s_t) \rightarrow \text{Execute} \ a_t \]

Hoque et al. 2021.
LazyDAGger

$s_t$  Query Robot Action $a_t = \pi_R(s_t)$  Execute $a_t$

Mode Select

$f(\text{Object})$

$\geq 0.5$ OR Mode = Sup

Query $a^H_t = \pi_H(s_t)$

Execute $\tilde{a}^H_t \sim \mathcal{N}(a^H_t, \sigma^2 I)$
LazyDAgger

Query Robot Action \( a_t = \pi_R(s_t) \)

Mode Select

\[ f(\cdot) \]

\[ \begin{align*} &\geq 0.5 \text{ OR Mode } = \text{ Sup} \\ &< \tau_R \text{ Set Mode } = \text{ Auto} \end{align*} \]

\[ \mathcal{L}(\cdot, \cdot) \]

Execute \( \tilde{a}_t^H \sim \mathcal{N}(a_t^H, \sigma^2 I) \)

\( s_t \)
Simulation Experiments

Context Switching Reduction

- 79%
- 56%
- 46%

Legend:
- Behavior Cloning
- DAgger
- SafeDAgger
- Ours
- Ours (- Noise)
- Ours (- Switch to Auto)
Simulation Experiments

Number of Context Switches

Learning Curves

Learning Curves: Ablations

- Behavior Cloning
- DAgger
- SafeDAgger
- Ours
- Ours (- Noise)
- Ours (- Switch to Auto)
Simulation Experiments

Hoque et al. 2021.
\( L = \frac{\text{(Time to perform one context switch)}}{\text{(Time to perform one action)}} \)
\[ L = \frac{\text{Time to perform one context switch}}{\text{Time to perform one action}} \]

\[ B(\pi) \triangleq C(\pi) \cdot (L + I(\pi)) \]
\[ L = \frac{\text{(Time to perform one context switch)}}{\text{(Time to perform one action)}} \]

\[ B(\pi) \triangleq C(\pi) \cdot (L + I(\pi)) \]

C = 20 switches
I = 10 actions
B = 20L + 100
\[ L = \frac{(\text{Time to perform one context switch})}{(\text{Time to perform one action})} \]

\[ B(\pi) \triangleq C(\pi) \cdot (L + I(\pi)) \]

\[ C = 20 \text{ switches} \quad C = 4 \text{ switches} \]
\[ I = 2 \text{ actions} \quad D = 20 \text{ actions} \]
\[ B = 20L + 40 \quad B = 4L + 80 \]
Define “cut-off latency” $L^* \geq 0$, as the minimum value such that $B(\text{SafeDAgger}) > B(\text{LazyDAgger})$ for all $L \geq L^*$.
Simulation Experiments

\[ L^* = 0.0 \]

\[ L^* = 4.3 \]

\[ L^* = 7.6 \]
(1) SMOOTH

(2) ALIGN

(3) FOLD
Limitations

• Parameter tuning

• Hard to know how many interventions will be requested.

• One human managing one robot.
When should a robot ask for help?

Novel (and risky)
When should a robot ask for help?

Novel (and risky)

Risky (but not novel)
Novelty Estimation

\[ m = 1 \]

\[ m = 2 \]

\[ m = 3 \]

Ensembling

in distribution
Novelty Estimation: Supervisor Mode
Risk Estimation

\[ Q_{\pi}^{\pi}(s_t, a_t) = \mathbb{E}_{\pi_r} \left[ \sum_{t' = t}^{\infty} \gamma^{t'-t} 1_{g(s'_t)}|s_t, a_t \right] \]

\( R = 1 \) if at goal, 0 otherwise

Assume Goal identifier

Expected prob of reaching goal / task success
Risk Estimation

\[ Q_{G}^{\pi_{r}}(s_t, a_t) = \mathbb{E}_{\pi_{r}} \left[ \sum_{t'=t}^{\infty} \gamma^{t'-t} 1_{G}(s'_t) \mid s_t, a_t \right] \]

\[ \text{Risk}^{\pi_{r}}(s, a) = 1 - \hat{Q}_{\phi, G}^{\pi_{r}}(s, a) \]
Risk Estimation

\[ Q_{G}^{\pi_{r}}(s_t, a_t) = \mathbb{E}_{\pi_{r}} \left[ \sum_{t' = t}^{\infty} \gamma^{t' - t} 1_{g}(s'_{t})|s_t, a_t \right] \]

\[ \text{Risk}^{\pi_{r}}(s, a) = 1 - \hat{Q}_{\phi, G}^{\pi_{r}}(s, a) \]

\[ J_{G}^{Q}(s_t, a_t, s_{t+1}; \phi) = \]

\[ \frac{1}{2} \left( \hat{Q}_{\phi, G}^{\pi_{r}}(s_t, a_t) - (1 - 1_{g}(s_t) + (1 - 1_{g}(s_t))\gamma \hat{Q}_{\phi, G}^{\pi_{r}}(s_{t+1}, \pi_{r}(s_{t+1}))) \right)^2 \]
Putting it all together…

\[
\text{Novelty}(s_t) > \delta_h \\
\text{OR} \\
\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h
\]

Switch to SUPERSVIS OR MODE
Putting it all together...

\[
\begin{align*}
\text{Novelty}(s_t) & > \delta_h \\
\text{OR} \\
\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) & > \beta_h
\end{align*}
\]

\[
\begin{align*}
||\pi_r(s_t) - \pi_h(s_t)||_2^2 & < \delta_r \\
\text{AND} \\
\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) & < \beta_r
\end{align*}
\]

Switch to SUPERVISOR MODE

\[
\begin{align*}
\text{Switch to} \\
\text{AUTONOMOUS MODE}
\end{align*}
\]
Putting it all together...

\[ \text{Novelty}(s_t) > \delta_h \quad \text{OR} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h \]

\[ ||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r \quad \text{AND} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r \]

Switch to SUPERVISOR MODE

Wait, didn’t we just double the number of hyperparameters?
Putting it all together...

**AUTONOMOUS MODE**

- **Novelty**\( (s_t) > \delta_h \)
- **Risk**\( ^\pi_r (s_t, \pi_r(s_t)) > \beta_h \)

Switch to **SUPERVISOR MODE**

**SUPERVISOR MODE**

- \( ||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r \)
- **Risk**\( ^\pi_r (s_t, \pi_r(s_t)) < \beta_r \)

Switch to **AUTONOMOUS MODE**

\[ \alpha = \frac{\text{desired}}{\text{# interventions}} \frac{\text{# robot actions}}{\text{# robot actions}} \]
Putting it all together...

**AUTONOMOUS MODE**

\[ \text{Novelty}(s_t) > \delta_h \quad \text{OR} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h \]

Switch to **SUPERVISOR MODE**

\[ ||\pi_r(s_t) - \pi_h(s_t)||^2_2 < \delta_r \quad \text{AND} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r \]

Switch to **AUTONOMOUS MODE**

\[ \alpha = \frac{\text{desired # interventions}}{\text{# robot actions}} \]

Set to 1-$\alpha$ quantiles of empirical data

Set to medians of empirical data
Putting it all together...

\[ \text{Novelty}(s_t) > \delta_h \quad \text{OR} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h \]

\[ ||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r \quad \text{AND} \quad \text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r \]

\[ \alpha = \frac{\text{desired}}{\text{desired}} \]

\[ \frac{\text{# interventions}}{\text{# robot actions}} \]

Switch to SUPervisory MODE

Switch to AUTonomouS MODE
ThriftyDAgger

Target percent of time human wants to give interventions.

ThriftyDAgger

ThriftyDAgger

Supervisor Mode (Risk)

Human Demonstration
Behavior Cloning

Behavior Cloning

ThriftyDAgger (autonomous)

User Study

N=10 subjects each control 3 robots in simulation.

Memory: Non-Match

Memory: Match

Robot-Gated

Human-Gated
ThriftyDAgger Qualitative Results

Survey Responses

Very High

Neutral

Very Low

Mental Demand

Frustration

SafeDAgger

HG-DAgger

LazyDAgger

ThriftyDAgger
User Study Quantitative Results

ThriftyDAgger had
• 21% fewer human interventions
• 57% more concentration pairs found
• 80% more throughput
Scalable and safe robot fleets are possible when robots ask for help in ways that minimize human supervisor burden.
Next time: Inverse Reinforcement Learning!