Introduction to Imitation Learning & the Behavior Cloning Algorithm
Recap

Infinite horizon Discounted MDPs

$$\mathcal{M} = \{S, A, \gamma, r, P, \mu\}$$

State visitation: $$d^\pi_h(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P^\pi_h(s; s_0)$$
Recap

Infinite horizon Discounted MDPs

\[ \mathcal{M} = \{ S, A, \gamma, r, P, \mu \} \]

State visitation: 
\[ d^\pi_\mu(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P^\pi_\mu(s; s_0) \]

Performance Difference Lemma:

What’s the perf difference between \( \pi \) & \( \pi' \)?
Recap

Infinite horizon Discounted MDPs

\[ \mathcal{M} = \{ S, A, \gamma, r, P, \mu \} \]

State visitation:

\[ d_\mu^\pi(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P_h(s; s_0) \]

Performance Difference Lemma:

What’s the perf difference between \( \pi \) & \( \pi' \)?

\[ V_\mu^\pi - V_\mu^{\pi'} = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_\mu^\pi} \left[ \mathbb{E}_{a \sim \pi(\cdot | s)} A_\mu^{\pi'}(s, a) \right] \]
Recap

Infinite horizon Discounted MDPs

\[ \mathcal{M} = \{ S, A, \gamma, r, P, \mu \} \]

State visitation:

\[ d^\pi_\mu(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P^\pi_h(s; s_0) \]

Performance Difference Lemma:

What’s the perf difference between \( \pi \) & \( \pi' \)?

\[
V^\pi_\mu - V^{\pi'}_\mu = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^\pi_\mu} \left[ \mathbb{E}_{a \sim \pi(\cdot|s)} A^{\pi'}(s, a) \right]
\]

The adv against \( \pi' \) averaged over the state distribution of \( \pi \)
Recap

Infinite horizon Discounted MDPs

\[ \mathcal{M} = \{S, A, \gamma, r, P, \mu\} \]

State visitation:

\[ d^\pi_\mu(s) = (1 - \gamma) \sum_{h=0}^{\infty} \gamma^h P^\pi_h(s; s_0) \]

Performance Difference Lemma:

What’s the perf difference between \( \pi \) & \( \pi' \)?

\[
V^\pi_\mu - V'^\pi_\mu = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^\pi_\mu} \left[ \mathbb{E}_{a \sim \pi(\cdot | s)} A^\pi'(s, a) \right]
\]

The adv against \( \pi' \) averaged over the state distribution of \( \pi \)
Outline for today:

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC
An Autonomous Land Vehicle In A Neural Network [Pomerleau, NIPS ‘88]
Imitation Learning
Imitation Learning
Imitation Learning
Imitation Learning

Expert Demonstrations
Imitation Learning

- SVM
- Gaussian Process
- Kernel Estimator
- Deep Networks
- Random Forests
- LWR
- ...

Expert Demonstrations

Machine Learning Algorithm
Imitation Learning

Expert Demonstrations

Machine Learning Algorithm

Policy $\pi$

- SVM
- Gaussian Process
- Kernel Estimator
- Deep Networks
- Random Forests
- LWR
- ...

Maps states to actions
Learning to Drive by Imitation

Input: Camera Image

Policy

Output: Steering Angle in [-1, 1]

[Pomerleau89, Saxena05, Ross11a]
Supervised Learning Approach: Behavior Cloning

[Widrow64, Pomerleau89]
Supervised Learning Approach: Behavior Cloning

[Expert Trajectories] → [Dataset]

[Images of steering wheels and a race track]

$X \rightarrow Y$
Supervised Learning Approach: Behavior Cloning

[Widrow64, Pomerleau89]
Supervised Learning Approach: Behavior Cloning

Learned Policy $\pi$

Mapping from state (image) to control (steering direction)

[Problems64,Pomerleau89]
But Poor Performance...
But Poor Performance...

Ready!
Outline

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC
Let's formalize the offline IL Setting and the Behavior Cloning algorithm

Discounted infinite horizon MDP $\mathcal{M} = \{S, A, \gamma, r, P, \rho, \pi^*\}$
Let's formalize the offline IL Setting and the Behavior Cloning algorithm.

Discounted infinite horizon MDP $\mathcal{M} = \{S, A, \gamma, r, P, \rho, \pi^*\}$

Ground truth reward $r(s, a) \in [0, 1]$ is unknown;
For simplicity, let's assume expert is a (nearly) optimal policy $\pi^*$
Let’s formalize the offline IL Setting and the Behavior Cloning algorithm.

Discounted infinite horizon MDP $\mathcal{M} = \{S, A, \gamma, r, P, \rho, \pi^*\}$

Ground truth reward $r(s, a) \in [0, 1]$ is unknown;
For simplicity, let’s assume expert is a (nearly) optimal policy $\pi^*$

We have a dataset $\mathcal{D} = (s_i^*, a_i^*)_{i=1}^M \sim d_{\pi^*}^\mu$
Let’s formalize the offline IL Setting and the Behavior Cloning algorithm

Discounted infinite horizon MDP $\mathcal{M} = \{S, A, \gamma, r, P, \rho, \pi^*\}$

Ground truth reward $r(s, a) \in [0, 1]$ is unknown;
For simplicity, let’s assume expert is a (nearly) optimal policy $\pi^*$

We have a dataset $\mathcal{D} = (s_i^*, a_i^*)_{i=1}^M \sim d^{\pi^*}$

Goal: learn a policy from $\mathcal{D}$ that is as good as the expert $\pi^*$
Let’s formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class \( \Pi = \{ \pi : S \mapsto \Delta(A) \} \)

BC is a Reduction to Supervised Learning:
Let’s formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class $\Pi = \{\pi : S \mapsto \Delta(A)\}$

BC is a Reduction to Supervised Learning:

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \mathcal{L}(\pi, s^*, a^*)$$

$\mathcal{L}$: loss function
Let’s formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class \( \Pi = \{ \pi : S \mapsto \Delta(A) \} \)

BC is a Reduction to Supervised Learning:

\[
\hat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell \left( \pi, s^*, a^* \right)
\]

Many choices of loss functions:
Let’s formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class \( \Pi = \{ \pi : S \mapsto \Delta(A) \} \)

BC is a Reduction to Supervised Learning:

\[ \hat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell (\pi, s^*, a^*) \]

Many choices of loss functions:

1. Negative log-likelihood (NLL):
\[ \ell (\pi, s^*, a^*) = - \ln \pi(a^* | s^*) \]
Let’s formalize the Behavior Cloning algorithm

BC Algorithm input: a restricted policy class \( \Pi = \{ \pi : S \mapsto \Delta(A) \} \)

BC is a Reduction to Supervised Learning:

\[
\hat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi, s^i, a^i)
\]

Many choices of loss functions:

1. Negative log-likelihood (NLL): \( \ell(\pi, s, a^*) = - \ln \pi(a^* | s^*) \)

2. Square loss (i.e., regression for continuous action): \( \ell(\pi, s^*, a^*) = ||\pi(s^*) - a^*||_2^2 \)
Analysis

Assumption: we are going to assume Supervised Learning succeeded
Assumption: we are going to assume Supervised Learning succeeded

$$\widehat{\pi} = \arg \min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi, s^*, a^*)$$

$$\mathbb{E}_{S \sim d_{\pi^*}^\mu} \left[ \mathbf{1} \left[ \widehat{\pi}(S) \neq \pi^*(S) \right] \right] \leq \epsilon \in \mathbb{R}^+$$
Analysis

Assumption: we are going to assume Supervised Learning succeeded

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \ell(\pi, s^*, a^*)$$

$$\mathbb{E}_{s \sim d^{\pi^*}_{\mu}} \mathbf{1} \left[ \hat{\pi}(s) \neq \pi^*(s) \right] \leq \epsilon \in \mathbb{R}^+$$

Note that here training and testing mismatch at this stage!
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$

$$\left(1 - \gamma\right)\left(V^{\pi^*} - V^{\pi^*}\right) = \mathbb{E}_{s \sim d_{\pi^*}} A(s, \pi^*(s))$$
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$

$$(1 - \gamma)(V^* - V^{\hat{\pi}}) = \mathbb{E}_{s \sim d^\pi} \Delta s \hat{\pi}(s, \pi^*(s))$$

$$= \mathbb{E}_{s \sim d^\pi} \Delta s \hat{\pi}(s, \pi^*(s)) - \mathbb{E}_{s \sim d^\pi} \Delta s \hat{\pi}(s, \hat{\pi}(s))$$

$$D - \frac{1}{1 - \gamma} \leq \hat{\pi}(s, a) = Q(s, a) - V^*(s) \leq \frac{1}{1 - \gamma}$$

$A^\pi(s, a) \in [-\frac{1}{1 - \gamma}, \frac{1}{1 - \gamma}]$, $\forall s, a$
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$

$$(1 - \gamma)(V^* - V^{\hat{\pi}}) = \mathbb{E}_{s \sim d^*} A^{\hat{\pi}}(s, \pi^*(s))$$
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\hat{\pi}$:

$$V^{\pi^*} - V^\hat{\pi} \leq \frac{2}{(1 - \gamma)^2} \epsilon$$

\[
(1 - \gamma)(V^\star - V^\hat{\pi}) = \mathbb{E}_{s \sim d^*} A^\hat{\pi}(s, \pi^*(s))
\]

\[
\leq \mathbb{E}_{s \sim d^*} \frac{2}{1 - \gamma} \mathbb{1} \{ \hat{\pi}(s) \neq \pi^*(s) \}
\]

\[
\leq \frac{2}{1 - \gamma} \epsilon \quad \Rightarrow \quad V^\star - V^\hat{\pi} \leq \frac{\epsilon}{(1 - \gamma)^2}
\]
Analysis

Theorem [BC Performance] With probability at least $1 - \delta$, BC returns a policy $\tilde{\pi}$:

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2}$$

The quadratic amplification is annoying

$$(1 - \gamma)(V^* - V^{\hat{\pi}}) = \mathbb{E}_{s \sim d^\pi} A^{\hat{\pi}}(s, \pi^*(s))$$

$$= \mathbb{E}_{s \sim d^\pi} A^{\hat{\pi}}(s, \pi^*(s)) - \mathbb{E}_{s \sim d^\pi} A^{\hat{\pi}}(s, \hat{\pi}(s))$$

$$\leq \mathbb{E}_{s \sim d^\pi} \frac{2}{1 - \gamma} \mathbf{1} \{ \hat{\pi}(s) \neq \pi^*(s) \}$$

$$\leq \frac{2}{1 - \gamma} \epsilon$$
Outline

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The distribution shift issue in BC
What could go wrong?

- Predictions affect future inputs/observations

[18]

[Pomerleau89, Daume09]

Learned Policy

Expert’s trajectory

We don’t have feedback from expert
Distribution Shift: Example

Initial state

Diagram:

- $s_0$ to $s_1$ with action $a_1$
- $s_1$ to $s_2$ with action $a_2$
- $s_2$ to $s_0$ with action $a_2$
Distribution Shift: Example

Initial state
Distribution Shift: Example

Initial state

\[ d_{s_0}^*(s_0) = 1 - \gamma, \quad d_{s_0}^*(s_1) = \gamma, \quad d_{s_0}^*(s_2) = 0 \]
Distribution Shift: Example

\[ r(s_1) = 1 \]

Initial state

\[ d^\pi_{s_0}(s_0) = 1 - \gamma, \quad d^\pi_{s_0}(s_1) = \gamma, \quad d^\pi_{s_0}(s_2) = 0 \]

\[
\begin{align*}
    d^\pi_{s_0}(s_0) &= (1-\delta) \sum_{h=0}^{\infty} \delta^h P^\pi_{1} (s_h; s_0) \\
    d^\pi_{s_0}(s_1) &= 1 - \gamma \\
    d^\pi_{s_0}(s_2) &= 0
\end{align*}
\]
Distribution Shift: Example

\[ r(s_1) = 1 \]

\[ d^\pi_{s_0}(s_0) = 1 - \gamma, \quad d^\pi_{s_0}(s_1) = \gamma, \quad d^\pi_{s_0}(s_2) = 0 \]

\[ V^\pi_{s_0} = \frac{\gamma}{1 - \gamma} = \left[ 0 + \gamma + \gamma^2 + \gamma^3 + \ldots \right] \]
Distribution Shift: Example

Assume SL returned such policy $\hat{\pi}$

$$\hat{\pi}(s_0) = \begin{cases} a_1 \text{ w/ prob } 1 - \epsilon/(1 - \gamma) \\ a_2 \text{ w/ prob } \epsilon/(1 - \gamma) \end{cases}, \quad \hat{\pi}(s_1) = a_2, \hat{\pi}(s_2) = a_2$$

$r(s_1) = 1$

Assume $\xi \rightarrow \text{small number}$

$V_{s_0}^\pi = \frac{\gamma}{1 - \gamma}$

$d_{s_0}^\pi(s_0) = 1 - \gamma$, $d_{s_0}^\pi(s_1) = \gamma$, $d_{s_0}^\pi(s_2) = 0$
Distribution Shift: Example

Assume SL returned such policy $\hat{\pi}$

$$\hat{\pi}(s_0) = \begin{cases} a_1 & \text{w/ prob } 1 - \epsilon/(1 - \gamma) \\ a_2 & \text{w/ prob } \epsilon/(1 - \gamma) \end{cases}, \quad \hat{\pi}(s_1) = a_2, \hat{\pi}(s_2) = a_2$$

We will have good supervised learning error:

$$\mathbb{E}_{s \sim d^\pi_{s_0}} \mathbb{E}_{a \sim \hat{\pi}(\cdot|s)} 1(a \neq \pi^*(s)) = \epsilon$$

at $s_0$: $\left(1 - \gamma \right) \cdot \frac{\epsilon}{1 - \gamma} + \gamma \cdot 0 + 0 = \epsilon$

Initial state

$r(s_1) = 1$

$V_{s_0}^\pi = \frac{\gamma}{1 - \gamma}$

$d^\pi_{s_0}(s_0) = 1 - \gamma, d^\pi_{s_0}(s_1) = \gamma, d^\pi_{s_0}(s_2) = 0$
Distribution Shift: Example

\[ V_{s_0} = \frac{\gamma}{1 - \gamma} \]

\[ V_{s_0}^* = \frac{\gamma}{1 - \gamma} - \frac{\epsilon \gamma}{(1 - \gamma)^2} = V_{s_0}^* - \frac{\epsilon \gamma}{(1 - \gamma)^2} \]

Assume SL returned such policy \( \hat{\pi} \)

\[ \hat{\pi}(s_0) = \begin{cases} 
\hat{\pi}(s_0) = a_1 & \text{w/ prob } 1 - \epsilon/(1 - \gamma) \\
\hat{\pi}(s_1) = a_2, \hat{\pi}(s_2) = a_2 & \text{w/ prob } \epsilon/(1 - \gamma) 
\end{cases} \]

We will have good supervised learning error:

\[ \mathbb{E}_{s \sim d^\pi} \mathbb{E}_{a \sim \hat{\pi}(\cdot|s)} 1(a \neq \pi^*(s)) = \epsilon \]

But we have quadratic error in performance:

\[ V_{s_0}^* = \frac{\gamma}{1 - \gamma} - \frac{\epsilon \gamma}{(1 - \gamma)^2} \]
Distribution Shift: Example

Assume SL returned such policy \( \hat{\pi} \)

\[
\hat{\pi}(s_0) = \begin{cases} 
  a_1 & \text{w/ prob } 1 - \epsilon/(1 - \gamma) \\
  a_2 & \text{w/ prob } \epsilon/(1 - \gamma)
\end{cases}, \quad \hat{\pi}(s_1) = a_2, \hat{\pi}(s_2) = a_2
\]

We will have good supervised learning error:

\[
\mathbb{E}_{s \sim d_{s_0}^\pi} \mathbb{E}_{a \sim \hat{\pi}(\cdot|s)} 1(a \neq \pi^*(s)) = \epsilon
\]

But we have quadratic error in performance:

\[
V_{s_0}^{\hat{\pi}} = \frac{\gamma}{1 - \gamma} - \frac{\epsilon \gamma}{(1 - \gamma)^2} = V_{s_0}^{\pi^*} - \frac{\epsilon \gamma}{(1 - \gamma)^2}
\]

Issue: once we make a mistake at \( s_0 \), we end up in \( s_2 \) which is not in the training data!
“If the network is not presented with sufficient variability in its training exemplars to cover the conditions it is likely to encounter…[it] will perform poorly”
An Autonomous Land Vehicle
In A Neural Network  [Pomerleau, NIPS ‘88]

“If the network is not presented with sufficient variability in its training exemplars to cover the conditions it is likely to encounter…[it] will perform poorly”
Summary for Today:

1. The most common imitation learning algorithm: BC

A reduction to supervised Learning, e.g., training classifier from $s^* \sim d_{\pi^*}^\mu, a^* = \pi^*(s^*)$
Summary for Today:

1. The most common imitation learning algorithm: BC

   A reduction to supervised Learning, e.g., training classifier from $s^* \sim d_{\mu^*}, a^* = \pi^*(s^*)$

2. Distribution shift:

   When execute the learned policy, we may deviate from the expert trajectories, causing compounding error
Summary for Today:

1. The most common imitation learning algorithm: BC

A reduction to supervised Learning, e.g., training classifier from $s^* \sim d_{\mu}^*, a^* = \pi^*(s^*)$

2. Distribution shift:

When execute the learned policy, we may deviate from the expert trajectories, causing compounding error

3. Again this demonstrates why RL/IL is harder than SL: we need to test our model on new data generated by our model