Imitation Learning: Behavior Cloning, Distribution Shift, & Distribution Matching

Sham Kakade and Wen Sun

CS 6789: Foundations of Reinforcement Learning
Announcements

HW3 is out last night and is due Nov 24th 11:59pm

(Lots of bonus questions, but try them out!)

No classes on Nov 17, 19, & 24 (university semi-final week)
Recap

Offline RL

\[ \mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^n, \text{ where } s_i, a_i \sim \mu, r_i = r(s_i, a_i), s'_i \sim P(\cdot | s_i, a_i) \]
Recap

Offline RL

\[ \mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^n, \text{ where } s_i, a_i \sim \mu, r_i = r(s_i, a_i), s'_i \sim P(\cdot | s_i, a_i) \]

Fitted Q Iteration: start from \( f_0 \in \mathcal{F} \)

\[
f_{t+1} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \left( f(s_i, a_i) - \left( r_i + \gamma \max_{a'} f_{t}(s'_i, a') \right) \right)^2_{\mathcal{F}f_{t}(s_i,a_i)}
\]
Recap

Offline RL

\[ \mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^n \], where \( s_i, a_i \sim \mu, r_i = r(s_i, a_i), s'_i \sim P(\cdot | s_i, a_i) \)

Fitted Q Iteration: start from \( f_0 \in \mathcal{F} \)

\[
f_{t+1} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \left( f(s_i, a_i) - \left( r_i + \gamma \max_{a'} f_t(s'_i, a') \right) \right)^2
\]

Performing regression from \((s_i, a_i)\) to \( \mathcal{F}_t(s_i, a_i) \)
Recap

Offline RL

\[ \mathcal{D} = \{ s_i, a_i, r_i, s'_i \}_{i=1}^n \], where \( s_i, a_i \sim \mu, r_i = r(s_i, a_i), s'_i \sim P( \cdot | s_i, a_i) \)

Fitted Q Iteration: start from \( f_0 \in \mathcal{F} \)

\[
    f_{t+1} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \left( f(s_i, a_i) - \left( r_i + \gamma \max_{a'} f_t(s'_i, a') \right) \right)^2
\]

Performing regression from \( (s_i, a_i) \) to \( \mathcal{T} f_t(s_i, a_i) \)

\[
    \sup_{\pi, s, a} \frac{d^\pi(s, a)}{\mu(s, a)} < \infty, \quad \forall f \in \mathcal{F}, \mathcal{T} f \in \mathcal{F}, \Rightarrow \text{FQI learns near-optimal policy in polynomially sample complexity}
\]
Today: Imitation Learning

1. Introduction of Imitation Learning

2. Offline Imitation Learning: Behavior Cloning

3. The hybrid Setting: Statistical Benefit and Distribution Matching
An Autonomous Land Vehicle In A Neural Network [Pomerleau, NIPS ‘88]
Imitation Learning
Imitation Learning
Imitation Learning
Imitation Learning

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

Expert Demonstrations
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

Output: Steering Angle in [-1, 1]

Policy

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

[Widrow64, Pomerleau89]

Expert Trajectories → Dataset
Supervised Learning Approach: Behavior Cloning

[Widrow64, Pomerleau89]
Supervised Learning Approach: Behavior Cloning

[Widrow64, Pomerleau89]
Supervised Learning Approach: Behavior Cloning

Expert Trajectories

Dataset

Learned Policy $\pi$

Mapping from state (image) to control (steering direction)
Ready!
But Poor Performance...
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; assume expert is a near 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; assume expert is a near optimal policy $\pi^*$

We have a dataset $\mathcal{D} = (s_i^*, a_i^* )_i^{M} \sim d^\pi^*$
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; assume expert is a near 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 with Maximum Likelihood Estimation (MLE):
Let’s formalize the Behavior Cloning algorithm

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

BC with Maximum Likelihood Estimation (MLE):

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^* | s_i^*)$$
Let’s formalize the Behavior Cloning algorithm

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

BC with Maximum Likelihood Estimation (MLE):

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a^*_i | s^*_i)$$

(We can reduce it to other supervised learning oracles such as classification, regression)
Analysis

Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^* | s_i^*)$$
Analysis

Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a^*_i | s^*_i)$$
Analysis

Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

Theorem [MLE Guarantee] With probability at least $1 - \delta$, we have:

$$\mathbb{E}_{s \sim d^{\pi^*}} \operatorname{tv} \left( \hat{\pi}(\cdot | s) - \pi^*(\cdot | s) \right) \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$
Analysis

Assumption: \( \Pi \) is discrete, and realizable, i.e., \( \pi^* \in \Pi \)

Step 1: Supervised learning (MLE) guarantee (see the book for reference to the classic MLE analysis):

Theorem [MLE Guarantee] With probability at least \( 1 - \delta \), we have:

\[
\mathbb{P}_{s \sim d^{\pi^*}} \left\| \frac{\hat{\pi}(\cdot | s) - \pi^*(\cdot | s)}{\Delta} \right\|_{tv} \leq \sqrt{\frac{\ln(\frac{1}{\delta})}{M}}
\]

This \( 1/\sqrt{M} \) rate should be expected:

no training and testing mismatch at this stage!
Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$
Analysis

Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

Step 2: Transfer supervised learning error to policy’s performance gap

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a^*_i \mid s^*_i)$$
Analysis

Assumption: $\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

Step 2: Transfer supervised learning error to policy’s performance gap

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

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\ln(|\Pi|/\delta)} \frac{\ln(|\Pi|/\delta)}{M}$$

MLE error

with supervised learning error

$$\mathbb{E}_{s \sim \pi^*(\cdot|s)} \left\| \hat{\pi}(\cdot|s) - \pi^*(\cdot|s) \right\|_{TV} \leq \sqrt{\ln(M)}$$
Analysis

Assumption: \( \Pi \) is discrete, and realizable, i.e., \( \pi^* \in \Pi \)

Step 2: Transfer supervised learning error to policy’s performance gap

Theorem [BC Sample Complexity] With probability at least \( 1 - \delta \), BC returns a policy \( \hat{\pi} \):

\[
V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M \text{ MLE error}}}.
\]

Note that \( 1/(1 - \gamma)^2 \) quadratic dependency on effective horizon
Analysis

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

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

where

$$\hat{\pi} = \arg \max \sum_{i=1}^{M} \ln \pi(a_i^* | s_i^*)$$

$M$ is the number of expert samples.
Analysis

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

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

MLE error

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

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

$$V^\pi - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

MLE error

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

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

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

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

MLE error

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

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

$$\leq \mathbb{E}_{s \sim d^s} \frac{1}{1 - \gamma} \| \pi^*(\cdot | s) - \hat{\pi}(\cdot | s) \|_1$$

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^* | s_i^*)$$
Analysis

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

$$V^{\pi^*} - V^{\hat{\pi}} \leq \frac{2}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$$

MLE error

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{i=1}^{M} \ln \pi(a_i^* | s_i^*)$$

$$\varepsilon \Rightarrow \frac{1}{(1 - \gamma)^2} \cdot \varepsilon$$

$$A^n = Q^n - V^n$$

Ideal $\frac{1}{1 - \gamma} \cdot \varepsilon$

Error-term from MLE
What could go wrong?

• Predictions affect future inputs/observations

[18]

[Pomerleau89,Daume09]
Distribution Shift: Intuitive Explanation

Let’s just focus on finite horizon (H) and deterministic policies here:

$$\mathbb{E}_{s \sim d_h^\pi} \hat{\pi}(s) \neq \pi^*(s) \leq \epsilon, \forall h$$
Distribution Shift: Intuitive Explanation

Let’s just focus on finite horizon (H) and deterministic policies here:

$$E_{s \sim d_h^*} \hat{\pi}(s) \neq \pi^*(s) \leq \epsilon, \forall h$$
Distribution Shift: Intuitive Explanation

Let’s just focus on finite horizon \((H)\) and deterministic policies here:

\[
\mathbb{E}_{s \sim d^*_h} \hat{\pi}(s) \neq \pi^*(s) \leq \epsilon, \forall h
\]
Distribution Shift: Intuitive Explanation

Let’s just focus on finite horizon ($H$) and deterministic policies here:

$$\mathbb{E}_{s \sim d^*_h} \hat{\pi}(s) \neq \pi^*(s) \leq \epsilon, \forall h$$
Distribution Shift: Intuitive Explanation

Let’s just focus on finite horizon ($H$) and deterministic policies here:

$$
\mathbb{E}_{s \sim d_{h}^{*}} \hat{\pi}(s) \neq \pi^{*}(s) \leq \epsilon, \forall h
$$
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”
A potential Fix
A potential Fix

Let’s roll out our policy in the real world, and compare our trajectories to the expert’s trajectories, and then refine our learned model.
The Hybrid Imitation Learning Setting:

Recall BC: we only use offline expert data—no interaction with the environment

Hybrid setting: offline expert data + simulator (e.g., known transition $P$)
The Hybrid Imitation Learning Setting:

Recall BC: we only use offline expert data—no interaction with the environment

Hybrid setting: offline expert data + simulator (e.g., known transition $P$)
Let’s formalize the Hybrid Setting

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

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

We have a dataset $\mathcal{D} = (s_i^*, a_i^*)_{i=1}^M \sim d^{\pi^*}$
Let’s formalize the Hybrid Setting

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

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

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

This time, we have a known transition $P$ (but we cannot plan because $r$ is unknown)
Let’s formalize the Hybrid Setting

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

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

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

This time, we have a known transition $P$ (but we cannot plan because $r$ is unknown)

**Key Q:** can we do better than offline IL Behavior Cloning (statistically at least—assuming infinite computation power)?
Key Idea: distribution matching

Integral probability metric (IPM)
Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions
Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathcal{F} = \{f : X \rightarrow \mathbb{R}\}$, and two distributions $p_1$ and $p_2$
Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathcal{F} = \{f : X \mapsto \mathbb{R}\}$, and two distributions $p_1$ and $p_2$

$$\text{IPM}_\mathcal{F}(p_1, p_2) = \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x) \right]$$
Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: $\mathcal{F} = \{ f : X \mapsto \mathbb{R} \}$, and two distributions $p_1$ and $p_2$

$$
\text{IPM}_\mathcal{F}(p_1, p_2) = \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x) \right]
$$

$$
\mathcal{F} = \{ f : \|f\|_\infty \leq 1 \} \Rightarrow \text{IPM}_\mathcal{F}(p_1, p_2) := \|p_1 - p_2\|_{tv}
$$
Key Idea: distribution matching

Integral probability metric (IPM)

Metric measures the divergence between two distributions

Given a discriminator class: \( \mathcal{F} = \{ f : X \mapsto \mathbb{R} \} \), and two distributions \( p_1 \) and \( p_2 \)

\[
\text{IPM}_{\mathcal{F}}(p_1, p_2) = \max_{f \in \mathcal{F}} \mathbb{E}_{x \sim p_1} f(x) - \mathbb{E}_{x \sim p_2} f(x)
\]

\( \mathcal{F} = \{ f : \| f \|_\infty \leq 1 \} \Rightarrow \text{IPM}_{\mathcal{F}}(p_1, p_2) := \| p_1 - p_2 \|_{tv} \)

\( \mathcal{F} = \{ f : f \text{ is 1-Lipschitz} \} \Rightarrow \text{IPM}_{\mathcal{F}}(p_1, p_2) := \text{wasserstein dis}(p_1, p_2) \)
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathcal{F} = \{ f : \|f\|_{\infty} \leq 1 \}$ (IPM corresponds to TV distance)
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: \( \mathcal{F} = \{ f : \| f \|_\infty \leq 1 \} \) (IPM corresponds to TV distance)

Same assumption as we had in BC:
\( \Pi \) is discrete, and realizable, i.e., \( \pi^* \in \Pi \)
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathcal{F} = \{ f : \| f \|_\infty \leq 1 \}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
$\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

**Step 1**: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall $P$ is known);
(This step is computationally inefficient)
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: \( \mathcal{F} = \{ f : \| f \|_{\infty} \leq 1 \} \) (IPM corresponds to TV distance)

Same assumption as we had in BC:
\( \Pi \) is discrete, and realizable, i.e., \( \pi^* \in \Pi \)

**Step 1:** for each \( \pi \in \Pi \), compute \( d^\pi \in \Delta(S \times A) \) (recall \( P \) is known);
(This step is computationally inefficient)

**Step 2:** select useful discriminators: for all pair \( \pi \& \pi' \), with \( \pi \neq \pi' 

\[
f_{\pi,\pi'} = \arg \max_{f \in \mathcal{F}} \left[ \Delta \right. \left. \sum_{s,a} P(s,a) \left( f(s,a) - \mathbb{E}_{s,a \sim d^\pi} f(s,a) \right) \right]
\]
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathcal{F} = \{ f : \| f \|_\infty \leq 1 \}$ (IPM corresponds to TV distance)

Same assumption as we had in BC: 
$\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

**Step 1:** for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall $P$ is known); 
(This step is computationally inefficient)

**Step 2:** select useful discriminators: for all pair $\pi \& \pi'$, with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\widetilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \& \pi' \in \Pi, \pi \neq \pi' \}$
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: $\mathcal{F} = \{f : \|f\|_\infty \leq 1\}$ (IPM corresponds to TV distance)

Same assumption as we had in BC:
$\Pi$ is discrete, and realizable, i.e., $\pi^* \in \Pi$

Step 1: for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall $P$ is known);
(This step is computationally inefficient)

Step 2: select useful discriminators: for all pair $\pi \& \pi'$, with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \arg \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s, a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s, a) \right]$$

Set refined discriminator class $\tilde{\mathcal{F}} := \{f_{\pi,\pi'} : \pi \& \pi' \in \Pi, \pi \neq \pi'\}$

Q: what is the size of $\tilde{\mathcal{F}}$?
Algorithm: Distribution Matching TV distance

Consider the Discriminator class: \( \mathcal{F} = \{ f : \|f\|_\infty \leq 1 \} \) (IPM corresponds to TV distance)

Same assumption as we had in BC:
\( \Pi \) is discrete, and realizable, i.e., \( \pi^* \in \Pi \)

**Step 1:** for each \( \pi \in \Pi \), compute \( d^\pi \in \Delta(S \times A) \) (recall \( P \) is known);
(This step is computationally inefficient)

**Step 2:** select useful discriminators: for all pair \( \pi \& \pi' \), with \( \pi \neq \pi' \)

\[
f_{\pi,\pi'} = \arg \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]
\]

Set refined discriminator class \( \tilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \& \pi' \in \Pi, \pi \neq \pi' \} \)

\[ \forall \pi \& \pi' \in \Pi, \|d^\pi - d^{\pi'}\|_{tv} = \max_{f \in \tilde{\mathcal{F}}} \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \]
**Algorithm: Distribution Matching TV distance**

**Step 1:** for each \( \pi \in \Pi \), compute \( d^\pi \in \Delta(S \times A) \) (recall \( P \) is known); (This step is computationally inefficient)

**Step2:** select useful discriminators: for all pair \( \pi \& \pi' \), with \( \pi \neq \pi' \)

\[
f_{\pi, \pi'} = \arg \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s, a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s, a) \right]
\]

Set refined discriminator class \( \mathcal{F}' := \{ f_{\pi, \pi'} : \pi \& \pi' \in \Pi, \pi \neq \pi' \} \)
Algorithm: Distribution Matching TV distance

**Step 1:** for each $\pi \in \Pi$, compute $d^\pi \in \Delta(S \times A)$ (recall $P$ is known); (This step is computationally inefficient)

**Step 2:** select useful discriminators: for all pair $\pi$ & $\pi'$, with $\pi \neq \pi'$

$$f_{\pi,\pi'} = \operatorname{arg\,max}_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^{\pi'}} f(s,a) \right]$$

Set refined discriminator class $\tilde{\mathcal{F}} := \{ f_{\pi,\pi'} : \pi \& \pi' \in \Pi, \pi \neq \pi' \}$

**Step 3:** Select a policy using expert dataset $\mathcal{D} = \{ s^*_i, a^*_i \}_{i=1}^M$
Algorithm: Distribution Matching TV distance

**Step 1:** for each $\pi \in \Pi$, compute $d^{\pi} \in \Delta(S \times A)$ (recall $P$ is known); (This step is computationally inefficient)

**Step 2:** select useful discriminators: for all pair $\pi, \pi'$, with $\pi \neq \pi'$

$$f_{\pi, \pi'} = \arg \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \mathbb{E}_{s,a \sim d^\pi} f(s,a) \right]$$

Set refined discriminator class $\mathcal{F} := \{ f_{\pi, \pi'} : \pi \neq \pi' \}$

**Step 3:** Select a policy using expert dataset $\mathcal{D} = \{ s^*_i, a^*_i \}_{i=1}^M$

$$\hat{\pi} := \arg \min_{\pi \in \Pi} \max_{f \in \mathcal{F}} \left[ \mathbb{E}_{s,a \sim d^\pi} f(s,a) - \frac{1}{M} \sum_{i=1}^M f(s^*_i, a^*_i) \right]$$
Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1 - \delta$, our algorithm finds a policy $\hat{\pi}$, s.t.,

$$V^{\pi^*} - V^{\hat{\pi}} \leq o \left( \frac{1}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$

suggesting error.
Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1 - \delta$, our algorithm finds a policy $\hat{\pi}$, s.t.,

$$V^{\pi^*} - \hat{V} \leq O\left( \frac{1}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$

1. Key step is to prove: $\|d^\hat{\pi} - d^{\pi^*}\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$
Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least \(1 - \delta\), our algorithm finds a policy \(\hat{\pi}\), s.t.,

\[
V_{\pi^*} - V_{\hat{\pi}} \leq O\left(\frac{1}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}\right)
\]

1. Key step is to prove:

\[
\|d_{\hat{\pi}} - d_{\pi^*}\|_{\text{tv}} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}
\]

2. For performance:

\[
V_{\pi^*} - V_{\pi} \leq \frac{1}{1 - \gamma} \left[\mathbb{E}_{s,a \sim d_{\pi^*}} r(s,a) - \mathbb{E}_{s,a \sim d_{\hat{\pi}}} r(s,a)\right]
\]

\[
\leq \max_{\pi \in \Pi} \|d_{\pi^*} - d_{\pi}\|_{\text{tv}}
\]
Theorem: Distribution Matching TV distance

Theorem [Dis-match w/ TV dist] With probability at least $1 - \delta$, our algorithm finds a policy $\hat{\pi}$, s.t.,

$$V^{\pi^*} - V^{\hat{\pi}} \leq \mathcal{O} \left( \frac{1}{1 - \gamma} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$

1. Key step is to prove: $\|d^{\hat{\pi}} - d^{\pi^*}\|_{tv} \leq \sqrt{\frac{\ln(|\Pi|/\delta)}{M}}$

2. For performance: $V^{\pi^*} - V^\pi \leq \frac{1}{1 - \gamma} \left[ \mathbb{E}_{s,a \sim d^{\pi^*}} r(s, a) - \mathbb{E}_{s,a \sim d^{\hat{\pi}}} r(s, a) \right]$

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

$$V^{\pi^*} - V^{\hat{\pi}} = \mathcal{O} \left( \frac{1}{(1 - \gamma)^2} \sqrt{\frac{\ln(|\Pi|/\delta)}{M}} \right)$$
A Policy's trajectory

Expert's trajectory
\( d^\pi \) : generator that generates state-action pairs

A Policy's trajectory

Expert's trajectory
$d^\pi$: generator that generates state-action pairs
d$\pi^*$: Ground truth state-action distribution (we have samples from it)
$d^\pi$ : generator that generates state-action pairs

$d^\pi^*$ : Ground truth state-action distribution (we have samples from it)

$\mathcal{F}$ : discriminators which distinguish red and blue

A Policy’s trajectory

Expert’s trajectory
Next lecture we will talk about a computationally efficient algorithm in the hybrid setting
Conclusion:

1. Offline RL: only use offline expert data
BC is simple and easy to implement, has reasonable guarantees; but the quadratic dependency on horizon could cause real problems
Conclusion:

1. **Offline RL:** only use offline expert data
   BC is **simple and easy to implement**, has reasonable guarantees; but the **quadratic dependency on horizon** could cause real problems

2. **Hybrid RL:** offline expert data + known transition (simulator)
   Statistically, Distribution-matching has **linear dependency on horizon**, but the algorithm is **computationally inefficient**

\[ \forall \pi \in \Pi, \exists \pi^* \in \Delta(S \times A) \]
Conclusion:

1. Offline RL: only use offline expert data
   BC is simple and easy to implement, has reasonable guarantees; but the quadratic dependency on horizon could cause real problems

2. Hybrid RL: offline expert data + known transition (simulator)
   Statistically, Distribution-matching has linear dependency on horizon, but the algorithm is computationally inefficient

Take home message:
There is a provable statistical benefit from the hybrid setting!
Ps: the distribution matching algorithm is very new (it was discovered when I was writing the book chapter…)