Imitation Learning
&
Behavioral Cloning

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning
Fall 2022

Today

- HW4 will be posted today. (Please start early!)
- Recap++
  - an example + Proximal Policy Optimization (PPO)
- Today: Theory
  1. ~~Overview~~ of PG, problems+successes
  2. Behavior Cloning

Recap++

Some Helpful Notation: Visitation Measures

$s_0 \sim \mu$.

- Visitation probability at time h: $\mathbb{P}_h(s_h, a_h \mid \mu, \pi)$
  (recall that we absorb h, into the state, i.e. $s \leftarrow (s, h)$)
- Average Visitation Measure:

$$d^{\pi}_{\mu}(s, a) = \frac{1}{H} \sum_{h=0}^{H-1} \mathbb{P}_h(s, a \mid \mu, \pi)$$

- With this def, we have:

$$J(\theta) := E_{s_0 \sim \mu_0}\left[V^{\pi_\theta}(s_0)\right]$$

$$= E\left[\sum_{h=0}^{H-1} r(s_h, a_h) \,\middle|\, \mu_0, \pi_\theta\right] = H \, E_{s,a \sim d^{\pi}_{\mu}}\left[r(s,a)\right]$$

$$= H \cdot \underline{\mathbb{E}_{s \sim d^{\pi_\theta}_{\mu}}} E_{a \sim \pi_\theta(s)}\left[r(s, a)\right]$$

TRPO

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\theta} H \cdot \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(s)} A^{\pi_{\theta_t}}(s, a) \right]$$

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta} \right) \leq \delta$$

We want to maximize local advantage against π_{θ_t}, but we want the new policy to be close to π_{θ_t} (in the KL sense)

How we can actually do the optimization here?
After all, we don't even know the analytical form of trajectory likelihood…

NPG derived from TRPO:

We did second-order Taylor expansion on the KL constraint, and we get:

$$\frac{1}{H} KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \approx \frac{1}{2}(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t)$$

$$F_{\theta_t} := \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}}\left[\nabla_\theta \ln \pi_{\theta_t}(a | s)\left(\nabla_\theta \ln \pi_{\theta_t}(a | s)\right)^\top\right] \in \mathbb{R}^{dim_\theta \times dim_\theta}$$

This leads to the following simplified constrained optimization:

$$\max_\theta \nabla_\theta J(\pi_{\theta_t})^\top \left(\theta - \theta_t\right)$$

s.t. $(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$

Algorithm: Natural Policy Gradient

Initialize θ_0

For t = 0, …

Estimate PG $\widehat{\nabla_\theta J(\pi_{\theta_t})}$

Estimate Fisher info-matrix $F_{\theta_t} := \widehat{\mathbb{E}}_{s,a\sim d_\mu^{\pi_{\theta_t}}} \nabla_\theta \ln \pi_{\theta_t}(a \mid s)\left(\nabla_\theta \ln \pi_{\theta_t}(a \mid s)\right)^\top$

Natural Gradient Ascent: $\theta_{t+1} = \theta_t + \eta \widehat{F}_{\theta_t}^{-1} \widehat{\nabla}_\theta J(\pi_{\theta_t})$

Using a tuned η or using $\eta = \sqrt{\dfrac{\delta}{\nabla_\theta J(\pi_{\theta_t})^\top F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})}}$

(We will implement it in HW4 on Cartpole)

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

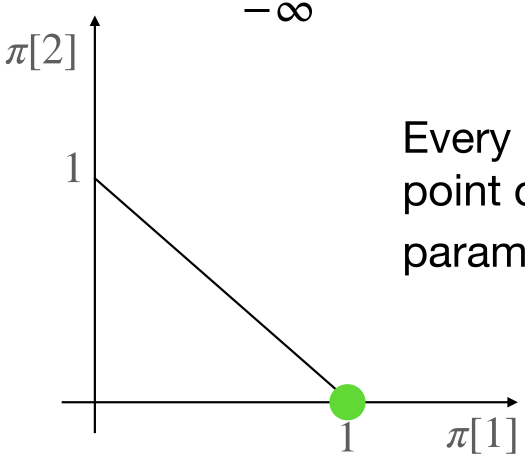
$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$

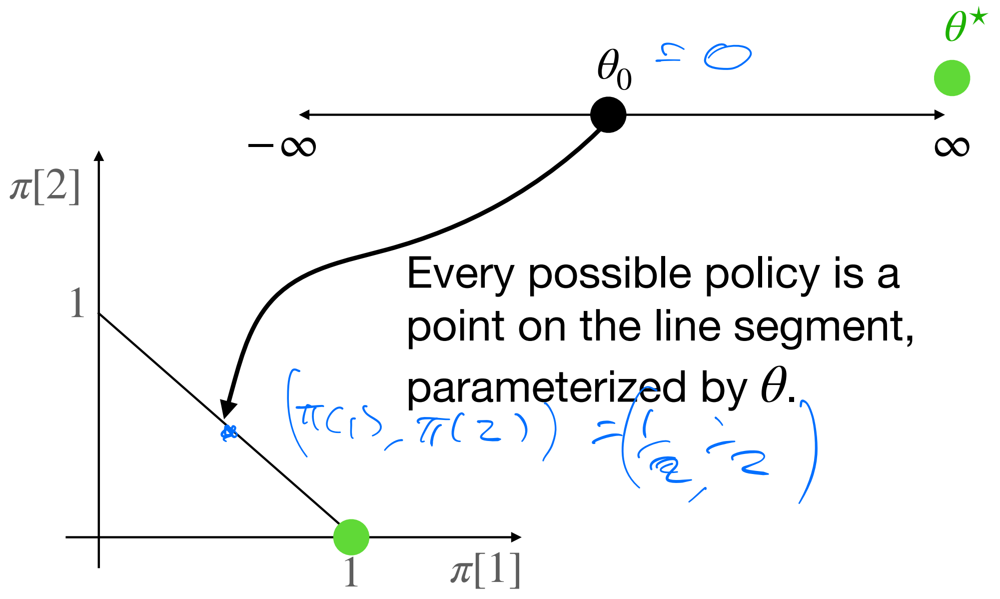


Every possible policy is a point on the line segment, parameterized by θ.

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Every possible policy is a point on the line segment, parameterized by θ.

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

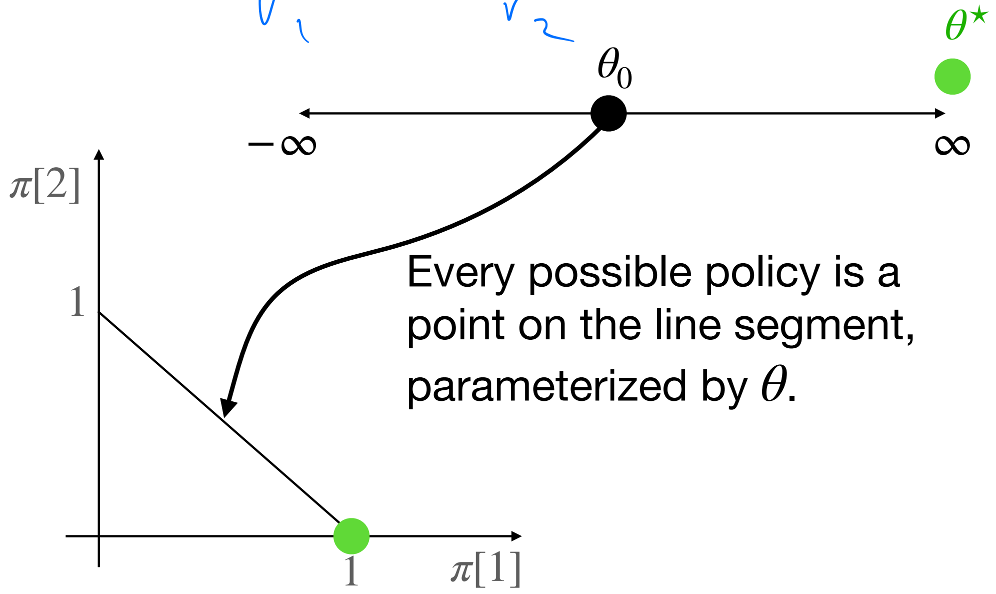
$\pi_\theta'(1) = \pi_\theta(1)(1-\pi_\theta(1))$

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)}\right)$$

$(r_1 - r_2) \cdot \pi_\theta(1)(1-\pi_\theta(1))$

Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

$$J(\theta) = \underset{r_1}{100} \cdot \pi_\theta[1] + \underset{r_2}{1} \cdot \pi_\theta[2]$$



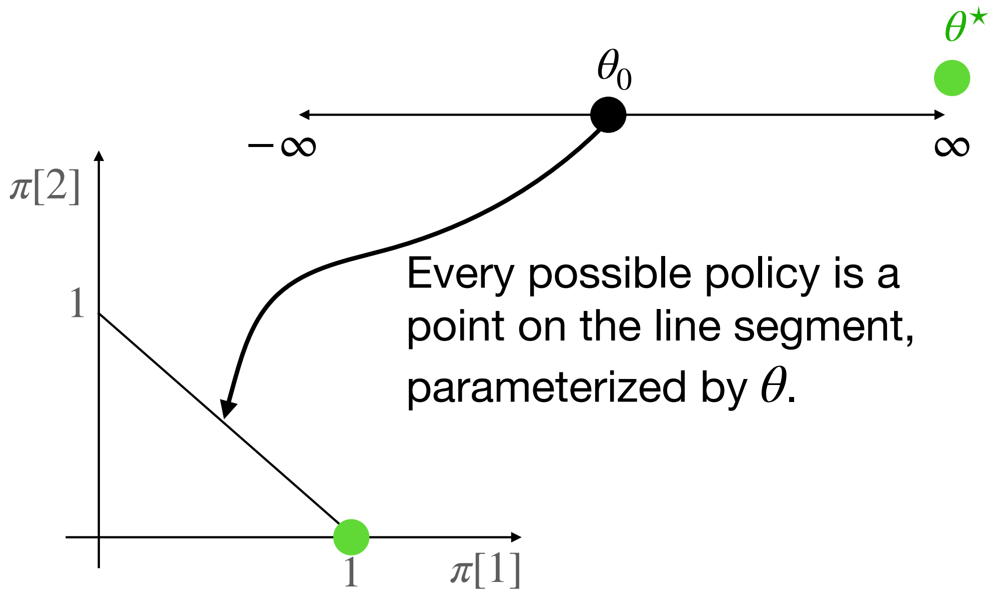
Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)}\right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$

Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$



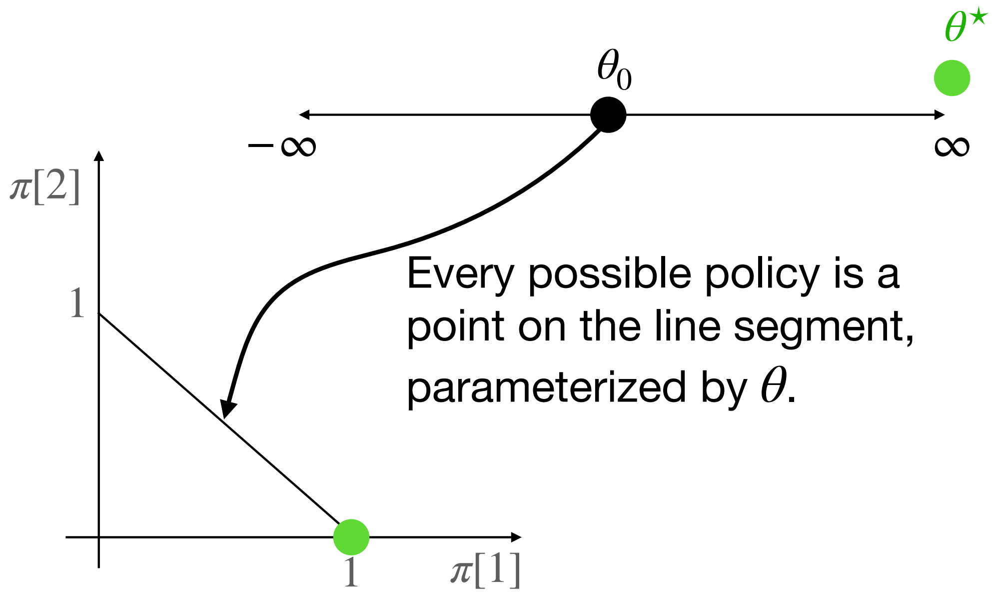
Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$

Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$

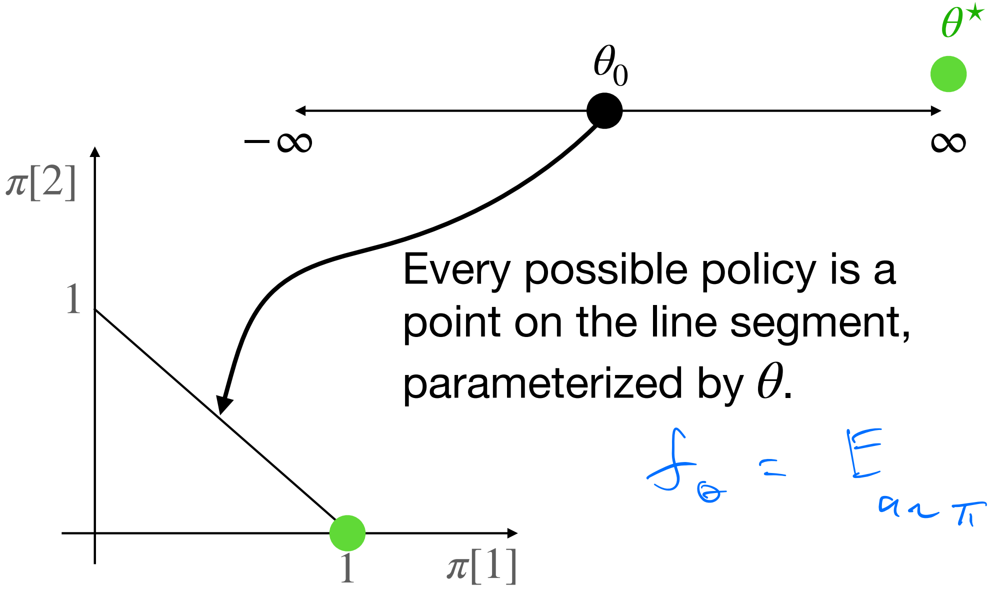


i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)}\right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Every possible policy is a point on the line segment, parameterized by θ.

$$f_\theta = E_{a\sim\pi}\left[\left(\left(\log \pi_\theta(a)\right)'\right)^2\right]$$

Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$

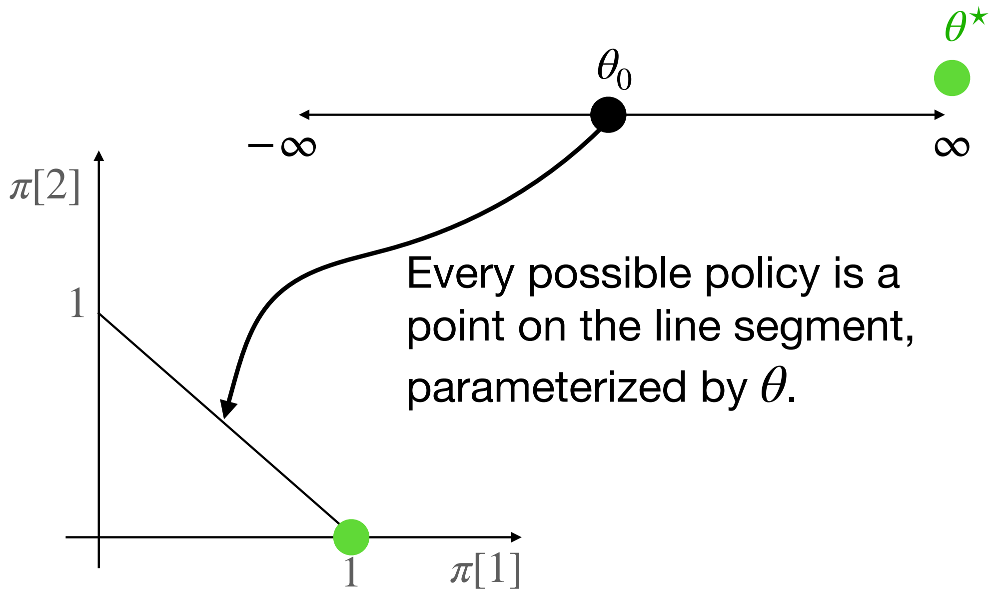
i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

Fisher information scalar: $f_\theta = \dfrac{\exp(\theta)}{(1+\exp(\theta))^2}$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)}\right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$

i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

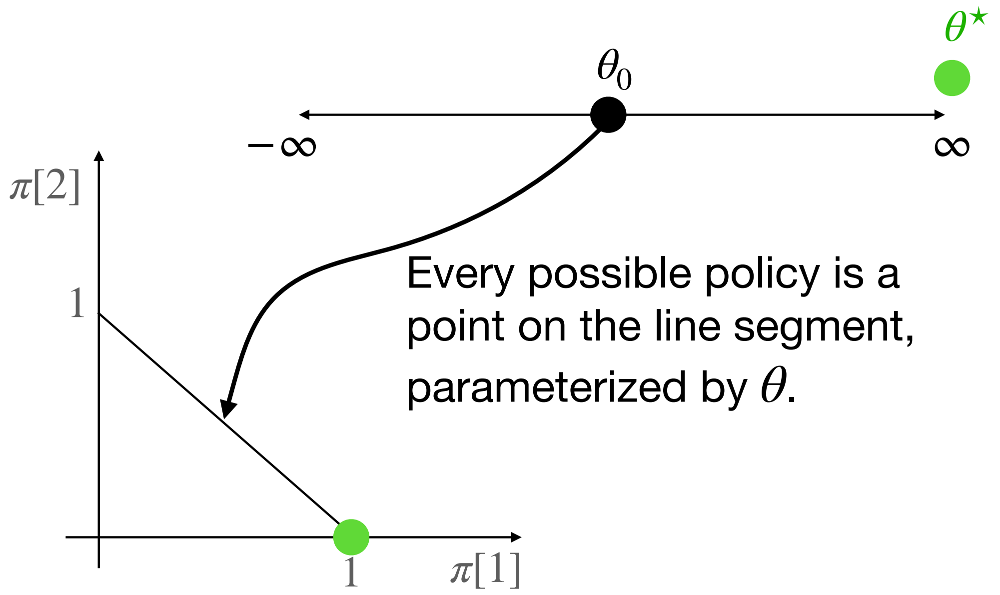
Fisher information scalar: $f_\theta = \dfrac{\exp(\theta)}{(1+\exp(\theta))^2}$

NPG: $\theta_{t+1} = \theta_t + \eta \dfrac{J'(\theta_t)}{f_{\theta_t}}$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$

i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

Fisher information scalar: $f_\theta = \dfrac{\exp(\theta)}{(1+\exp(\theta))^2}$

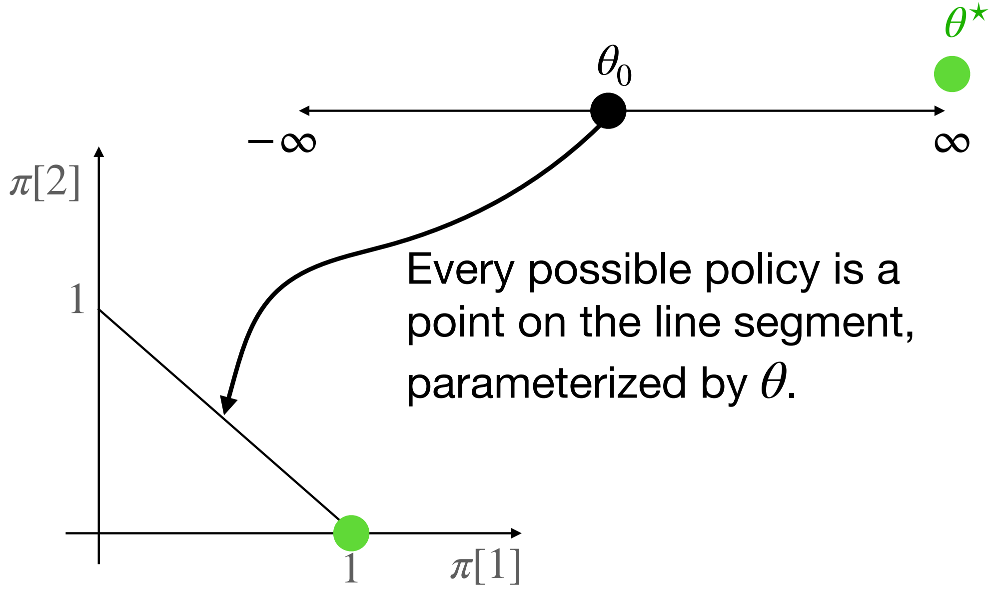
NPG: $\theta_{t+1} = \theta_t + \eta \dfrac{J'(\theta_t)}{f_{\theta_t}} = \theta_t + \eta \cdot 99$

$(r_1 - r_2)$

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Every possible policy is a point on the line segment, parameterized by θ.

Gradient: $J'(\theta) = \dfrac{99 \exp(\theta)}{(1 + \exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99 \exp(\theta_t)}{(1 + \exp(\theta_t))^2}$

i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

Fisher information scalar: $f_\theta = \dfrac{\exp(\theta)}{(1 + \exp(\theta))^2}$

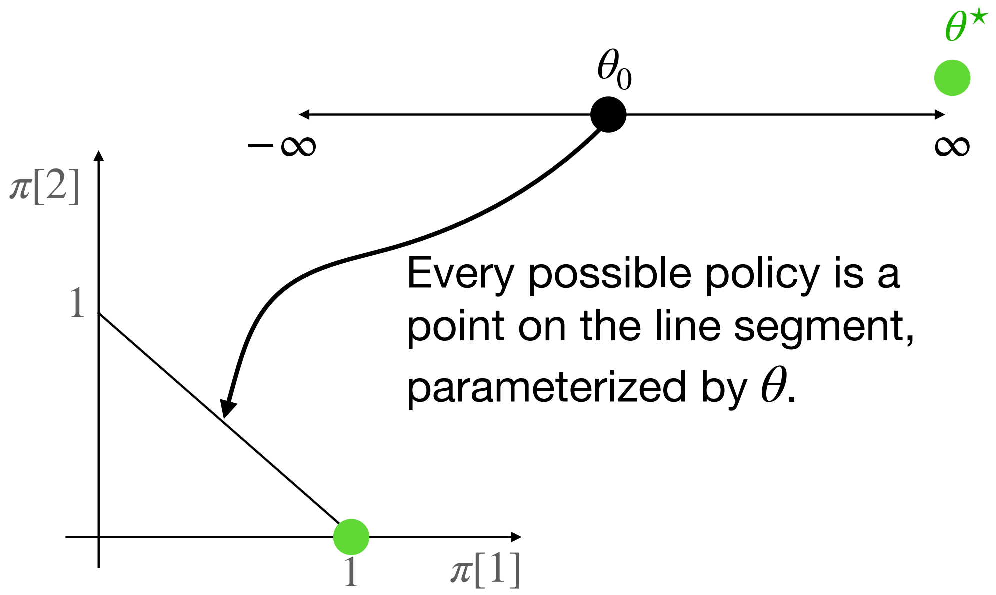
NPG: $\theta_{t+1} = \theta_t + \eta \dfrac{J'(\theta_t)}{f_{\theta_t}} = \theta_t + \eta \cdot 99$

NPG moves to $\theta = \infty$ much more quickly (for a fixed η)

Example of Natural Gradient on 1-d problem: 2 actions, 1 state

$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$J(\theta) = 100 \cdot \pi_\theta[1] + 1 \cdot \pi_\theta[2]$$



Every possible policy is a point on the line segment, parameterized by θ.

Gradient: $J'(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta_{t+1} = \theta_t + \eta \dfrac{99\exp(\theta_t)}{(1+\exp(\theta_t))^2}$

i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $J'(\theta) \to 0$ as $\theta \to \infty$

Fisher information scalar: $f_\theta = \dfrac{\exp(\theta)}{(1+\exp(\theta))^2}$

NPG: $\theta_{t+1} = \theta_t + \eta \dfrac{J'(\theta_t)}{f_{\theta_t}} = \theta_t + \eta \cdot 99$

NPG moves to $\theta = \infty$ much more quickly (for a fixed η)

Proximal Policy Optimization (PPO):
A computationally fast extension of NPG:

Given an current policy π^t, we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_{\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s, a) \right]$$

Proximal Policy Optimization (PPO): A computationally fast extension of NPG:

Given an current policy π^t, we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a\,|\,s)\,|\,\pi_\theta(a\,|\,s)\right)\right]}_{\text{regularization}}$$

Proximal Policy Optimization (PPO): A computationally fast extension of NPG:

Given an current policy π^t, we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a) \right] \underbrace{- \lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}} \left[\text{KL}\left(\pi_{\theta_t}(a\,|\,s) \,|\, \pi_\theta(a\,|\,s) \right) \right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

Proximal Policy Optimization (PPO): A computationally fast extension of NPG:

Given an current policy π^t, we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a\,|\,s)\,|\,\pi_\theta(a\,|\,s)\right)\right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)} \frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_t}(a\,|\,s)} A^{\pi_{\theta_t}}(s,a)\right] - \lambda \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\left[-\ln \pi_\theta(a\,|\,s)\right]$$

Proximal Policy Optimization (PPO):
A computationally fast extension of NPG:

Given an current policy π^t, we perform policy update to π^{t+1}

Proximal Policy Optimization (PPO)

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a\,|\,s)\,|\,\pi_\theta(a\,|\,s)\right)\right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_t}(a\,|\,s)} A^{\pi_{\theta_t}}(s,a)\right] - \lambda \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\left[-\ln \pi_\theta(a\,|\,s)\right]$$

PPO: Perform a few steps of mini-batch SGA on $\ell(\theta)$ to approximate $\arg\max_\theta \ell(\theta)$

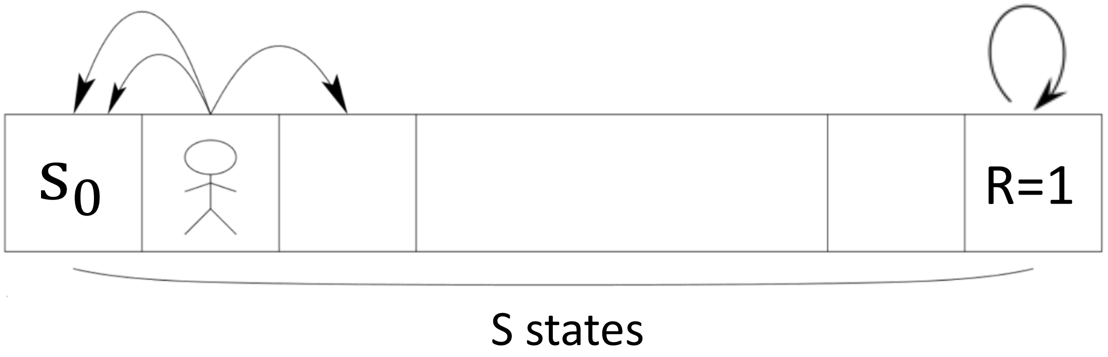
Today:

Optimality in Markov Decision Processes

Outline:

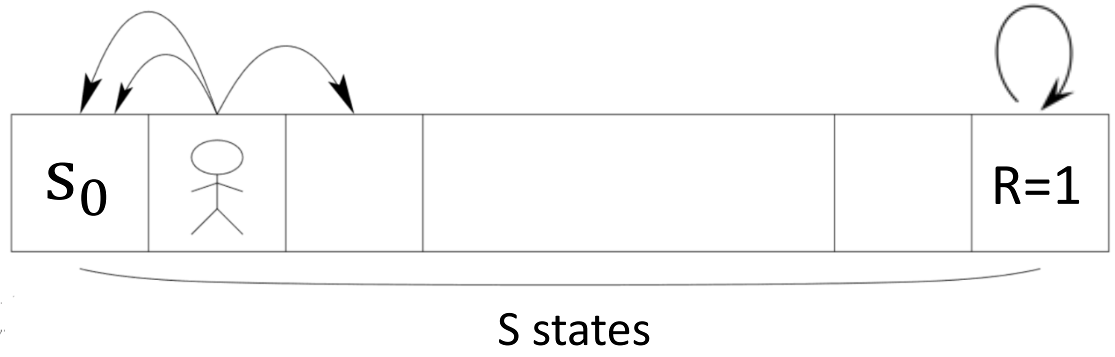
1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

“Lack of Exploration” leads to Optimization and Statistical Challenges



Thrun ‘92

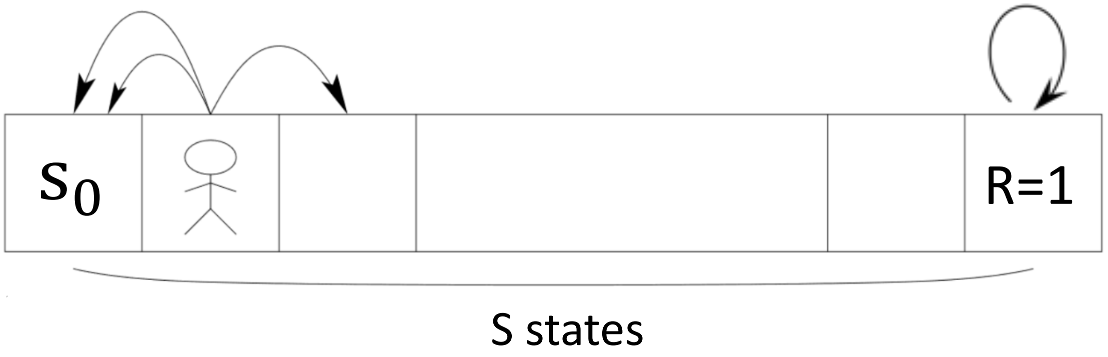
“Lack of Exploration” leads to Optimization and Statistical Challenges



Thrun ‘92

- Suppose $|S| \approx H$ or $|S| \approx 1/(1-\gamma)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).

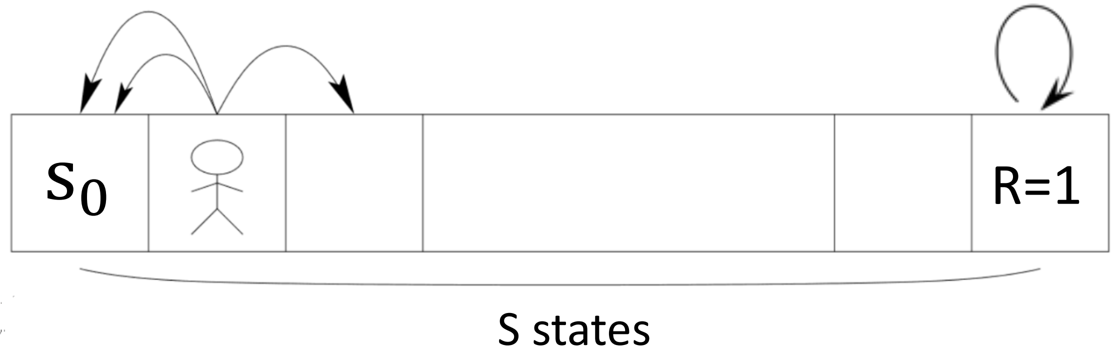
"Lack of Exploration" leads to Optimization and Statistical Challenges



Thrun '92

- Suppose $|S| \approx H$ or $|S| \approx 1/(1-\gamma)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).
- A randomly initialized policy has prob. $O(1/3^{|S|})$ of hitting the goal state in a single trajectory.

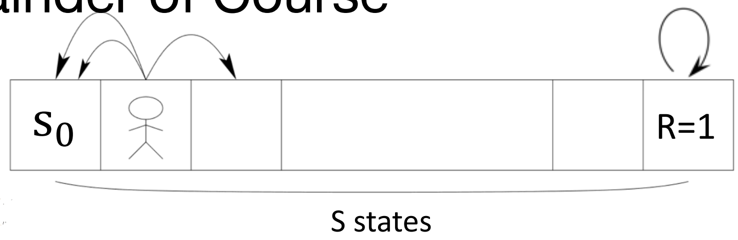
“Lack of Exploration” leads to Optimization and Statistical Challenges



Thrun ‘92

- Suppose $|S| \approx H$ or $|S| \approx 1/(1-\gamma)$ & $\mu(s_0) = 1$ (i.e. we start at s_0).
- A randomly initialized policy has prob. $O(1/3^{|S|})$ of hitting the goal state in a single trajectory.
- Implications:
  - Any sample based policy iteration approach (starting with this policy) requires $O(3^{|S|})$ trajectories to make progress at the very first step.
  - Same for any sample based PG method.
  - Related: even if we had exact gradients, the “landscape” is such that these gradients are exponentially small, at randomly initialized policy (see AJKS Ch 11).

Implications/Comments/Remainder of Course

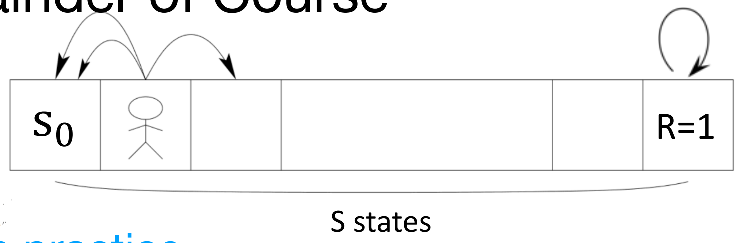


Thrun '92

Implications/Comments/Remainder of Course

- Sometimes exploration is (or can be made) "easier" in practice
  - Random strategies can reach "rewarding milestones"
  - We can design/"shape" the reward function to help us out.

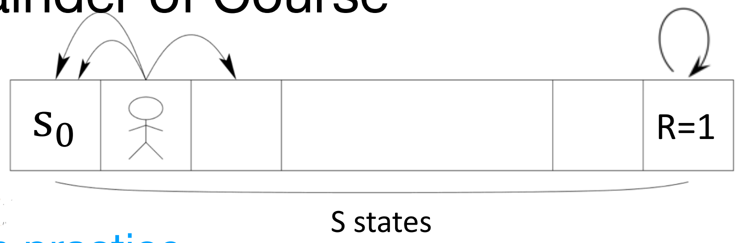
Implications/Comments/Remainder of Course



Thrun '92

- Sometimes exploration is (or can be made) "easier" in practice
  - Random strategies can reach "rewarding milestones"
  - We can design/"shape" the reward function to help us out.
- We can try to make the distribution μ to have better coverage.
  - For small problems, μ being uniform would make all these issues go away. (for large problems, μ being uniform may not help at all. Why?)
  - Ideally, μ having support on where a good policy tends to visit is helpful (sometimes we can't design μ)

Implications/Comments/Remainder of Course



Thrun '92

- Sometimes exploration is (or can be made) "easier" in practice
  - Random strategies can reach "rewarding milestones"
  - We can design/"shape" the reward function to help us out.
- We can try to make the distribution μ to have better coverage.
  - For small problems, μ being uniform would make all these issues go away. (for large problems, μ being uniform may not help at all. Why?)
  - Ideally, μ having support on where a good policy tends to visit is helpful (sometimes we can't design μ)
- Course:
  - A little theory with regards to μ and PG. (today) PG has better guarantees than approx DP methods (in terms of μ).
  - Imitation learning (starting today). An expert gives us samples from a "good" μ.
  - Explicit Exploration: for the "tabular case" (we will mix UCB with VI!)

Outline:

1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

Let's compare fitted DP and PG for "Linear" Parameterizations of Q-functions and Policies

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and
parameter $\theta \in \mathbb{R}^d$

1. Linear Functions

$$f_\theta(s, a) = \theta^\top \phi(s, a)$$

1. Softmax linear Policy

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

Fitted Policy Improvement Guarantees (optional)

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0 \sim \mu}[Q^\pi(s, a)]$
  (the theory is better suited to this. See AJKS).

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0 \sim \mu}[Q^\pi(s,a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \le \delta$, and $\min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \le \delta_\infty$

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0, a_0 \sim \mu}[Q^\pi(s, a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \le \delta$, and $\min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \le \delta_\infty$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)
    δ_∞: the worse case error (often unreasonable to expect to be small)

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0 \sim \mu}[Q^\pi(s,a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \le \delta, \text{ and } \quad \min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \le \delta_\infty$$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)
    δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0 \sim \mu}[Q^\pi(s,a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \le \delta, \text{ and } \quad \min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \le \delta_\infty$$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)
    δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

- Suppose that we use a # samples that is poly in d & $1/\epsilon_{stat}$ for both fittedPI and NPG.

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0, a_0 \sim \mu}[Q^\pi(s, a)]$ (the theory is better suited to this. See AJKS).

$\|f\|_\infty = \max_{s,a} f(s,a)$

- Approximation error: For all policies, suppose that for all π,

$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \le \delta$, and $\min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \le \delta_\infty$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)

    δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

- Suppose that we use a # samples that is poly in d & $1/\epsilon_{stat}$ for both fittedPI and NPG.
- FittedPI will return a policy π^{FPI} with the performance guarantee:

$$J(\pi^{FPI}) \ge J(\pi^\star) - \epsilon_{stat} - 2H^2\delta_\infty$$

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0\sim\mu}[Q^\pi(s,a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top\phi(s,a)\right)^2\right] \leq \delta, \text{ and } \min_\theta \|Q^\pi - \theta^\top\phi\|_\infty \leq \delta_\infty$$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)

    δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

- Suppose that we use a # samples that is poly in d & $1/\epsilon_{stat}$ for both fittedPI and NPG.
- FittedPI will return a policy π^{FPI} with the performance guarantee:

$$J(\pi^{FPI}) \geq J(\pi^\star) - \epsilon_{stat} - 2H^2\delta_\infty$$

- NPG has the same guarantee.

Fitted Policy Improvement Guarantees (optional)

- Let $s_0, a_0 \sim \mu$ now be the starting "state-action" distribution. $J(\pi) = E_{s_0,a_0\sim\mu}[Q^\pi(s,a)]$ (the theory is better suited to this. See AJKS).
- Approximation error: For all policies, suppose that for all π,

$$\min_\theta E_{s,a\sim\mu}\left[\left(Q^\pi(s,a) - \theta^\top \phi(s,a)\right)^2\right] \leq \delta, \text{ and } \quad \min_\theta \|Q^\pi - \theta^\top \phi\|_\infty \leq \delta_\infty$$

  - δ: the average case supervised learning error (reasonable to expect this can be made small)
    δ_∞: the worse case error (often unreasonable to expect to be small)

[Theorem:] (informal, see AJKS Ch 4+13)

- Suppose that we use a # samples that is poly in d & $1/\epsilon_{stat}$ for both fittedPI and NPG.
- FittedPI will return a policy π^{FPI} with the performance guarantee:

$$J(\pi^{FPI}) \geq J(\pi^\star) - \epsilon_{stat} - 2H^2\delta_\infty$$

- NPG has the same guarantee.
- NPG also has a stronger guarantee: Suppose μ has "reasonable support" on where $\pi^\star$ tends to visit, i.e. suppose:

$$\max_{s,a}\left(\frac{d_\mu^{\pi^\star}(s,a)}{\mu(s,a)}\right) \leq C$$

  then NPG will return a policy with sub-optimality determined by C and the average case error δ:

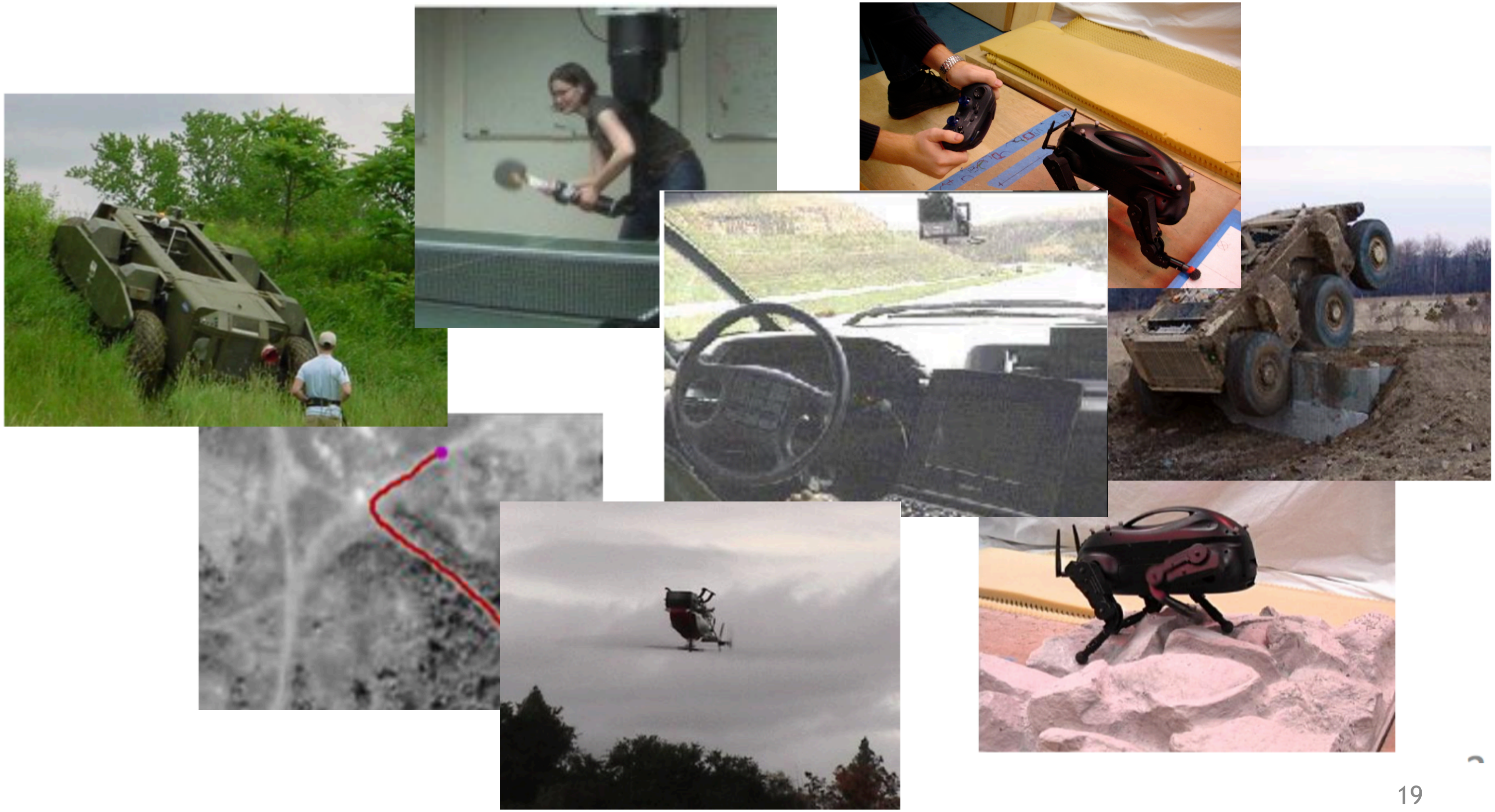
$$J(\pi^{NPG}) \geq J(\pi^\star) - \epsilon_{stat} - 2H^2 C\delta$$

Outline:

1. Exploration and the starting measure μ
2. Theory: (natural) policy gradients vs fitted Dynamic programming
3. Behavioral Cloning

3a. Introduction of Imitation Learning

Imitation Learning



Imitation Learning

Imitation Learning

Imitation Learning

Expert Demonstrations

Imitation Learning

Expert Demonstrations

Machine Learning Algorithm

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

Imitation Learning

Expert Demonstrations → Machine Learning Algorithm → Policy π

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

Maps *states* to actions

Learning to Drive by Imitation

[Pomerleau89, Saxena05, Ross11a]

Input:

Camera Image

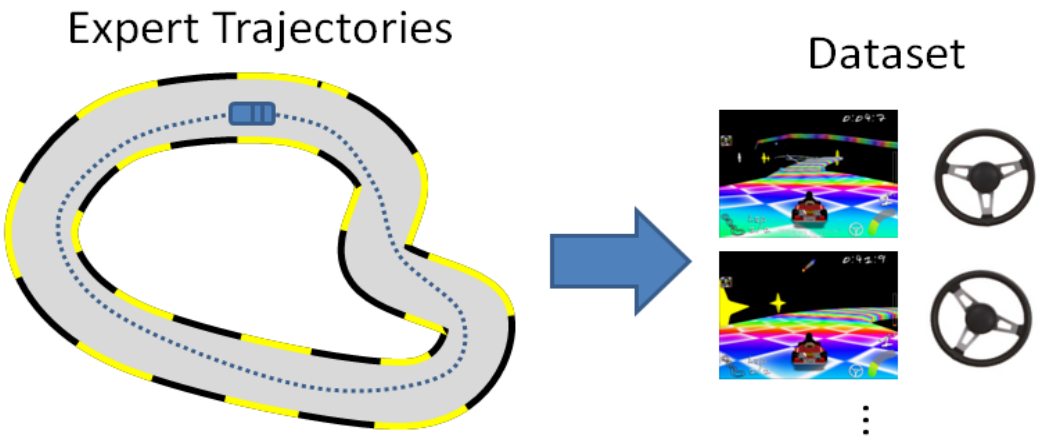
Policy

Output:

Steering Angle
in [-1, 1]

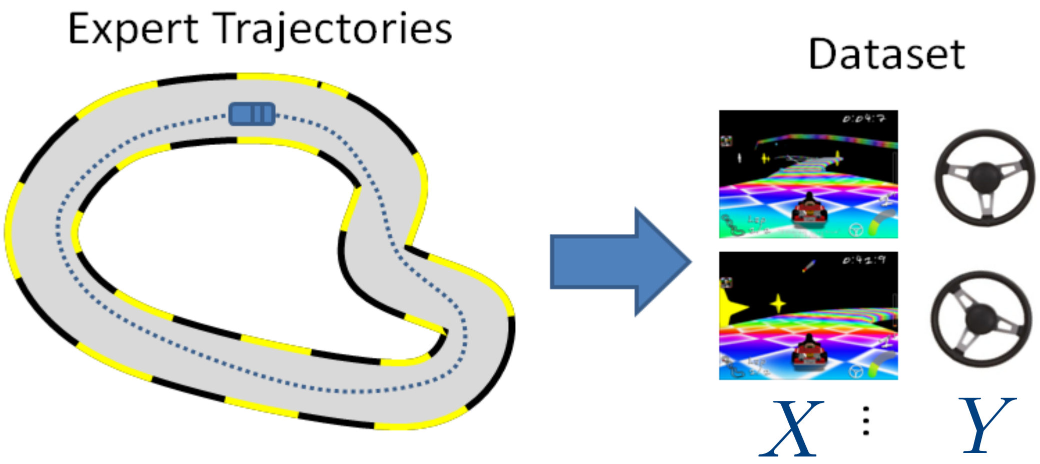
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



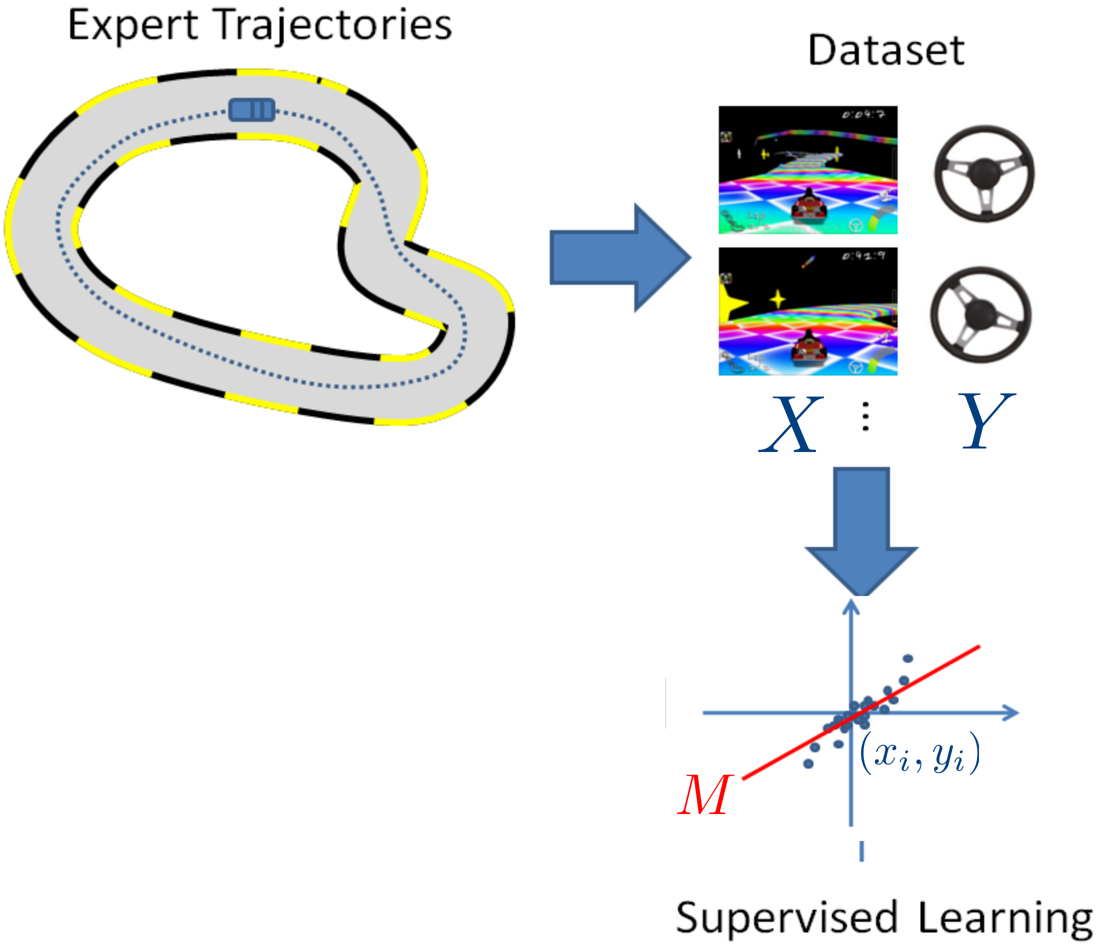
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



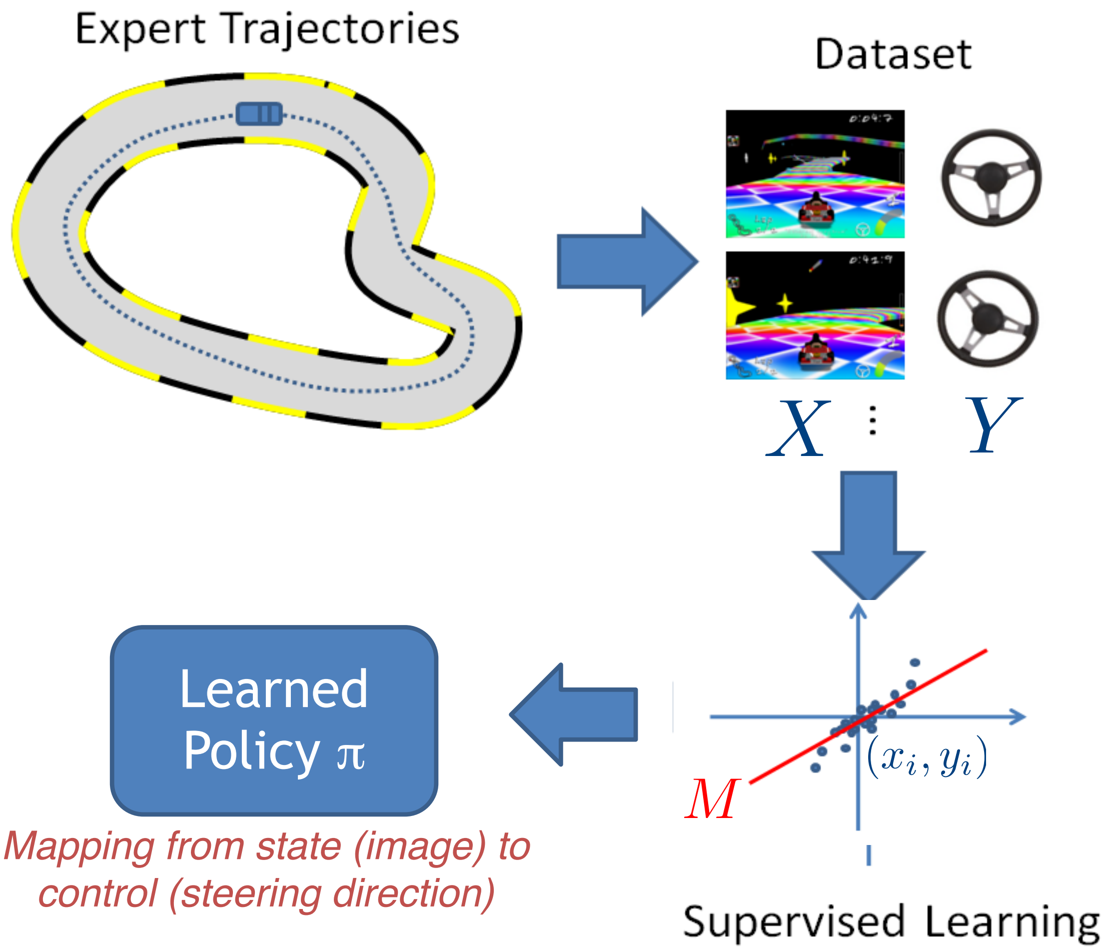
[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



[Widrow64,Pomerleau89]

Supervised Learning Approach: Behavior Cloning



3b. Offline Imitation Learning: Behavior Cloning

Let's formalize the offline IL Setting and the Behavior Cloning algorithm

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

Let’s formalize the offline IL Setting and the Behavior Cloning algorithm

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

Ground truth reward $r(s, a) \in [0,1]$ is unknown;

For simplicity, let’s assume expert is a (nearly) optimal policy $\pi^\star$

Let’s formalize the offline IL Setting and the Behavior Cloning algorithm

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

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

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

Let's formalize the offline IL Setting and the Behavior Cloning algorithm

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

Ground truth reward $r(s, a) \in [0,1]$ is unknown;

For simplicity, let's assume expert is a (nearly) optimal policy $\pi^\star$

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

Goal: learn a policy from $\mathscr{D}$ that is as good as the expert $\pi^\star$

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:

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

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:

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

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:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{i=1}^{M} \ell\left(\pi, s_i^\star, a_i^\star\right)$$

$\pi(s_i^*) \approx a_i^*$

Many choices of loss functions:

1. Negative log-likelihood (NLL): $\ell(\pi, s, a^\star) = -\ln \pi(a^\star \mid s^\star)$

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:

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

Many choices of loss functions:

1. Negative log-likelihood (NLL): $\ell(\pi, s, a^\star) = -\ln \pi(a^\star \mid s^\star)$

2. square loss (i.e., regression for continuous action): $\ell(\pi, s, a^\star) = \|\vec{\pi}(s) - \vec{a}^\star\|_2^2$

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

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

Analysis

Assumption: we are going to assume Supervised Learning succeeded

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

Note that here training and testing mismatch at this stage!

Analysis

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

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

$$\leq 2H^2 \varepsilon$$

Analysis

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

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

The quadratic amplification is annoying

Summary:

1. TRPO/NPG/PPO
2. Exploration/μ/Guarantees
3. Behavioral Cloning

1-minute feedback form: https://bit.ly/3RHtIxy