From TRPO/NPG to Proximal Policy Optimization (PPO)

Lucas Janson

CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Today

- Feedback from last lecture
- Recap
- TRPO -> NPG derivation
- Proximal Policy Optimization (PPO)
- Importance sampling

Feedback from feedback forms

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Today

- Feedback from last lecture
- Recap
- TRPO -> NPG derivation
- Proximal Policy Optimization (PPO)
- Importance sampling

PG with a Learned Baseline:

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\left(R_h(\tau) - b(s_h, h)\right)$

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\left(R_h(\tau) - b(s_h, h)\right)$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \,|\, s_h) \big(R_h(\tau) - b(s_h, h) \big)$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(R_h(\tau) - b(s_h, h) \right)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
    $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\left(R_h(\tau) - b(s_h, h)\right)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
    2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Compute $g'(\theta^k, \tau, \widetilde{b}())$

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h)\big(R_h(\tau) - b(s_h, h)\big)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
    2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Compute $g'(\theta^k, \tau, \widetilde{b}())$
    3. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau, \widetilde{b}())$

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\big(R_h(\tau) - b(s_h, h)\big)$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
    2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Compute $g'(\theta^k, \tau, \widetilde{b}())$
    3. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau, \widetilde{b}())$

Note that regardless of our choice of $\widetilde{b}$, we still get unbiased gradient estimates.

The Performance Difference Lemma (PDL)

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

$$V^{\widetilde{\pi}}(s) - V^{\pi}(s) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\pi},s}} \left[\sum_{h=0}^{H-1} A^{\pi}(s_h, a_h, h) \right]$$

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

$$V^{\widetilde{\pi}}(s) - V^{\pi}(s) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\pi},s}} \left[\sum_{h=0}^{H-1} A^{\pi}(s_h, a_h, h) \right]$$

Comments:

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

$$V^{\widetilde{\pi}}(s) - V^{\pi}(s) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\pi},s}} \left[\sum_{h=0}^{H-1} A^{\pi}(s_h, a_h, h) \right]$$

Comments:

- Helps us think about error analysis, instabilities of fitted PI, and sub-optimality.

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

$$V^{\widetilde{\pi}}(s) - V^{\pi}(s) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\pi},s}} \left[\sum_{h=0}^{H-1} A^{\pi}(s_h, a_h, h) \right]$$

Comments:

- Helps us think about error analysis, instabilities of fitted PI, and sub-optimality.
- Helps to understand algorithm design (TRPO, NPG, PPO)

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under $\widetilde{\pi}$. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

$$V^{\widetilde{\pi}}(s) - V^{\pi}(s) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\pi},s}} \left[\sum_{h=0}^{H-1} A^{\pi}(s_h, a_h, h) \right]$$

Comments:

- Helps us think about error analysis, instabilities of fitted PI, and sub-optimality.
- Helps to understand algorithm design (TRPO, NPG, PPO)
- This also motivates the use of “local” methods (e.g. policy gradient descent)

Back to Fitted Policy Iteration

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h)\right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h)\right]$

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h)\right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h)\right]$
- In particular, $\hat{A}^{\pi^k}$ should be close to A^{π^k} where π^k visits often…

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h) \right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h) \right]$
- In particular, $\hat{A}^{\pi^k}$ should be close to A^{π^k} where π^k visits often…
- But it could be very bad in places π^k visits rarely, and nothing stops π^{k+1} from visiting those bad places very often!

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h) \right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h) \right]$
- In particular, $\hat{A}^{\pi^k}$ should be close to A^{π^k} where π^k visits often…
- But it could be very bad in places π^k visits rarely, and nothing stops π^{k+1} from visiting those bad places very often!
- So π^{k+1} could end up being (much) worse than π^k

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h)\right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}}\left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h)\right]$
- In particular, $\hat{A}^{\pi^k}$ should be close to A^{π^k} where π^k visits often…
- But it could be very bad in places π^k visits rarely, and nothing stops π^{k+1} from visiting those bad places very often!
- So π^{k+1} could end up being (much) worse than π^k

- Problem is a mismatch between expectations: what we really want is

$$\mathbb{E}_{\tau \sim \rho_{\pi^{k+1},s}}\left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h)\right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^{k+1},s}}\left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h)\right]$$

Back to Fitted Policy Iteration

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- In Fitted Policy Iteration, $\hat{A}^{\pi^k} \approx A^{\pi^k}$ is achieved via supervised learning on $\tau_1, \ldots \tau_N \sim \rho_{\pi^k}$
- This means we expect $\mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h) \right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^k,s}} \left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h) \right]$
- In particular, $\hat{A}^{\pi^k}$ should be close to A^{π^k} where π^k visits often…
- But it could be very bad in places π^k visits rarely, and nothing stops π^{k+1} from visiting those bad places very often!
- So π^{k+1} could end up being (much) worse than π^k

- Problem is a mismatch between expectations: what we really want is

$$\mathbb{E}_{\tau \sim \rho_{\pi^{k+1},s}} \left[\sum_{h=0}^{H-1} \hat{A}^{\pi^k}(s_h, a_h, h) \right] \approx \mathbb{E}_{\tau \sim \rho_{\pi^{k+1},s}} \left[\sum_{h=0}^{H-1} A^{\pi^k}(s_h, a_h, h) \right]$$

- One way to ensure this: keep $\pi^{k+1} \approx \pi^k$

Trust Region Policy Optimization (TRPO)

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   try to approximately solve:
   $$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1} \sim \rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h \sim \pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$
   $$\text{s.t. } KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \leq \delta$$
3. Return π_{θ^K}

- We want to maximize local advantage against π_{θ^k}, but we want the new policy to be close to π_{θ^k} (in the KL sense)
- How do we implement this with sampled trajectories?)

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P \mid Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P \mid Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P \mid Q) = KL(Q \mid P) = 0$

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P|Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P|Q) = KL(Q|P) = 0$

If $P = \mathcal{N}(\mu_1, \sigma^2 I), Q = \mathcal{N}(\mu_2, \sigma^2 I)$, then $KL(P|Q) = \dfrac{1}{2\sigma^2}\|\mu_1 - \mu_2\|^2$

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P \,|\, Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P \,|\, Q) = KL(Q \,|\, P) = 0$

If $P = \mathcal{N}(\mu_1, \sigma^2 I), Q = \mathcal{N}(\mu_2, \sigma^2 I)$, then $KL(P \,|\, Q) = \frac{1}{2\sigma^2} \|\mu_1 - \mu_2\|^2$

Fact:

$KL(P \,|\, Q) \geq 0$, and is 0 if and only if $P = Q$

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$

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

Intuition: maximize local advantage
subject to being incremental (in KL)

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right] \longrightarrow \text{First-order Taylor expansion at } \theta^k$$

$$\text{s.t. } KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \leq \delta \longrightarrow \text{second-order Taylor expansion at } \theta^k$$

Intuition: maximize local advantage
subject to being incremental (in KL)

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

→ First-order Taylor expansion at θ^k

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

→ second-order Taylor expansion at θ^k

Intuition: maximize local advantage subject to being incremental (in KL)

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$

→ First-order Taylor expansion at θ^k

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

→ second-order Taylor expansion at θ^k

Intuition: maximize local advantage subject to being incremental (in KL)

$$\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$

→ First-order Taylor expansion at θ^k

s.t. $KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \leq \delta$

→ second-order Taylor expansion at θ^k

Intuition: maximize local advantage subject to being incremental (in KL)

$$\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

TRPO is locally equivalent to a much simpler algorithm

TRPO at iteration k:

$$\max_\theta \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

→ First-order Taylor expansion at θ^k

s.t. $KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \le \delta$

→ second-order Taylor expansion at θ^k

Intuition: maximize local advantage subject to being incremental (in KL)

$$\max_\theta \nabla_\theta J(\theta^k)^\top(\theta-\theta^k)$$

s.t. $(\theta-\theta^k)^\top F_{\theta^k}(\theta-\theta^k) \le \delta$

(Where F_{θ^k} is the "Fisher Information Matrix")

Natural Policy Gradient (NPG): A "leading order" equivalent program to TRPO:

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

3. Return π_{θ^K}

Natural Policy Gradient (NPG): A "leading order" equivalent program to TRPO:

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

3. Return π_{θ^K}

- Where $\nabla_\theta J(\theta^k)$ is the gradient of $J(\theta)$ evaluated at θ^k, and
- F_θ is (basically) the Fisher information matrix at $\theta \in \mathbb{R}^d$, defined as:

$$F_\theta := \mathbb{E}_{\tau \sim \rho_{\pi_\theta}} \left[\nabla_\theta \ln \rho_\theta(\tau) \left(\nabla_\theta \ln \rho_\theta(\tau) \right)^\top \right] \in \mathbb{R}^{d \times d}$$

$$= \mathbb{E}_{\tau \sim \rho_{\pi_\theta}} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) \left(\nabla_\theta \ln \pi_\theta(a_h | s_h) \right)^\top \right]$$

NPG has a closed form update!

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

3. Return π_{θ^K}

NPG has a closed form update!

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

3. Return π_{θ^K}

Linear objective and quadratic convex constraint: we can solve it optimally!

NPG has a closed form update!

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

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

3. Return π_{θ^K}

Linear objective and quadratic convex constraint: we can solve it optimally!

Indeed this gives us:

$$\theta^{k+1} = \theta^k + \eta F_{\theta^k}^{-1} \nabla_\theta J(\theta^k)$$

NPG has a closed form update!

1. Initialize θ^0
2. For $k = 0,\ldots,K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\theta^k)^\top (\theta - \theta^k)$$

s.t. $(\theta - \theta^k)^\top F_{\theta^k} (\theta - \theta^k) \le \delta$

3. Return π_{θ^K}

Linear objective and quadratic convex constraint: we can solve it optimally!

Indeed this gives us:

$$\theta^{k+1} = \theta^k + \eta F_{\theta^k}^{-1} \nabla_\theta J(\theta^k)$$

Where $\eta = \sqrt{\dfrac{\delta}{\nabla_\theta J(\theta^k)^\top F_{\theta^k}^{-1} \nabla_\theta J(\theta^k)}}$

An Implementation: Sample Based NPG

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   - Obtain approximation of Policy Gradient: $\hat{g} \approx \nabla_\theta J(\theta^k)$
   - Obtain approximation of Fisher information: $\hat{F} \approx F_{\theta^k}$
   - Natural Gradient Ascent: $\theta^{k+1} = \theta^k + \eta \hat{F}^{-1} \hat{g}$
3. Return π_{θ^K}

An Implementation: Sample Based NPG

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   - Obtain approximation of Policy Gradient: $\hat{g} \approx \nabla_\theta J(\theta^k)$
   - Obtain approximation of Fisher information: $\hat{F} \approx F_{\theta^k}$
   - Natural Gradient Ascent: $\theta^{k+1} = \theta^k + \eta \hat{F}^{-1} \hat{g}$
3. Return π_{θ^K}

(We will implement it in HW4 on Cartpole)

Today

- Feedback from last lecture
- Recap
- TRPO -> NPG derivation
- Proximal Policy Optimization (PPO)
- Importance sampling

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta}(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{ constant } + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{ constant } + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta - \theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{constant} + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta\mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}\right]$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{ constant } + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta\mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}\right]$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta}(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{ constant } + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta\mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta}(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\mathbb{E}_{a_h\sim\pi_{\theta}(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}\right]$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{constant} + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta\mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}\right]$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k} = \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)R_h(\tau)\right]\Bigg|_{\theta=\theta^k}$$

First Order Expansion on the Objective Function

$$f^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\approx f^k(\theta^k) + (\theta-\theta^k)\cdot\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k} = \text{ constant } + (\theta-\theta^k)\cdot\underbrace{\nabla_\theta f^k(\theta)\,|_{\theta=\theta^k}}$$

$$= \nabla_\theta\mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k}\right]$$

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right]\Bigg|_{\theta=\theta^k}$$

$$= \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\Bigg|_{\theta=\theta^k} = \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\nabla_\theta\ln\pi_\theta(a_h\,|\,s_h)R_h(\tau)\right]\Bigg|_{\theta=\theta^k} = \nabla_\theta J(\theta)\,|_{\theta=\theta^k}$$

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\tilde{\theta}} | \rho_{\theta})$$

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\widetilde{\theta}} | \rho_\theta)$$

$$\ell(\theta) \approx \ell(\widetilde{\theta}) + (\theta - \widetilde{\theta})^\top \nabla_\theta \ell(\theta)|_{\theta=\widetilde{\theta}} + \frac{1}{2}(\theta - \widetilde{\theta})^\top \left[\nabla_\theta^2 \ell(\theta)|_{\theta=\widetilde{\theta}} \right] (\theta - \widetilde{\theta})$$

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\tilde{\theta}} | \rho_\theta)$$

$$\ell(\theta) \approx \ell(\tilde{\theta}) + (\theta - \tilde{\theta})^\top \nabla_\theta \ell(\theta)|_{\theta=\tilde{\theta}} + \frac{1}{2}(\theta - \tilde{\theta})^\top \left[\nabla_\theta^2 \ell(\theta)|_{\theta=\tilde{\theta}}\right](\theta - \tilde{\theta})$$

$$\ell(\tilde{\theta}) = KL(\rho_{\tilde{\theta}} | \rho_{\tilde{\theta}}) = 0$$

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\tilde{\theta}} | \rho_{\theta})$$

$$\ell(\theta) \approx \ell(\tilde{\theta}) + (\theta - \tilde{\theta})^\top \nabla_\theta \ell(\theta)|_{\theta=\tilde{\theta}} + \frac{1}{2}(\theta - \tilde{\theta})^\top \left[\nabla_\theta^2 \ell(\theta)|_{\theta=\tilde{\theta}}\right](\theta - \tilde{\theta})$$

$$\ell(\tilde{\theta}) = KL(\rho_{\tilde{\theta}} | \rho_{\tilde{\theta}}) = 0$$

We will show that $\nabla_\theta \ell(\theta)|_{\theta=\tilde{\theta}} = 0$, and $\nabla_\theta^2 \ell(\theta)|_{\theta=\tilde{\theta}}$ has the claimed form!

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\widetilde{\theta}} | \rho_\theta) \qquad (\rho_{\widetilde{\theta}} := \rho_{\pi_{\theta k}} \text{ and } \rho_\theta := \rho_{\pi_\theta})$$

$$\ell(\theta) \approx \ell(\widetilde{\theta}) + (\theta - \widetilde{\theta})^\top \nabla_\theta \ell(\theta)|_{\theta=\widetilde{\theta}} + \frac{1}{2}(\theta - \widetilde{\theta})^\top \left[\nabla_\theta^2 \ell(\theta)|_{\theta=\widetilde{\theta}}\right](\theta - \widetilde{\theta})$$

$$\ell(\widetilde{\theta}) = KL(\rho_{\widetilde{\theta}} | \rho_{\widetilde{\theta}}) = 0$$

We will show that $\nabla_\theta \ell(\theta)|_{\theta=\widetilde{\theta}} = 0$, and $\nabla_\theta^2 \ell(\theta)|_{\theta=\widetilde{\theta}}$ has the claimed form!

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_{\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_{\theta}(\tau)}\right] = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_{\theta}(\tau)\right]$$

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\nabla_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\nabla_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln\frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln\rho_{\widetilde{\theta}}(\tau) - \ln\rho_\theta(\tau)\right]$$

$$\nabla_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\nabla_\theta \ln\rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

$$= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\frac{\nabla_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)}\Big|_{\theta=\widetilde{\theta}}$$

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\begin{aligned}\nabla_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} &= -\,\mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\nabla_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}} \\ &= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\frac{\nabla_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)}\Big|_{\theta=\widetilde{\theta}} \\ &= -\sum_\tau \nabla_\theta \rho_\theta(\tau)\Big|_{\theta=\widetilde{\theta}}\end{aligned}$$

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\nabla_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\nabla_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

$$= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\frac{\nabla_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)}\Big|_{\theta=\widetilde{\theta}}$$

$$= -\sum_\tau \nabla_\theta \rho_\theta(\tau)\Big|_{\theta=\widetilde{\theta}} = -\nabla_\theta \sum_\tau \rho_\theta(\tau)\Big|_{\theta=\widetilde{\theta}}$$

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\tilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln \frac{\rho_{\tilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln \rho_{\tilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\nabla_\theta \ell(\theta)\Big|_{\theta=\tilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\nabla_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\tilde{\theta}}$$

$$= -\sum_\tau \rho_{\tilde{\theta}}(\tau)\frac{\nabla_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)}\Big|_{\theta=\tilde{\theta}}$$

$$= -\sum_\tau \nabla_\theta \rho_\theta(\tau)\Big|_{\theta=\tilde{\theta}} = -\nabla_\theta \sum_\tau \rho_\theta(\tau)\Big|_{\theta=\tilde{\theta}} \quad = 0$$

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\tilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau \sim \rho_{\tilde{\theta}}}\left[\ln \frac{\rho_{\tilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau \sim \rho_{\tilde{\theta}}}\left[\ln \rho_{\tilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\,\mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\nabla^2_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln\frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\ln\rho_{\widetilde{\theta}}(\tau) - \ln\rho_\theta(\tau)\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\widetilde{\theta}}}\left[\nabla^2_\theta \ln\rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

$$= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\left(\frac{\nabla^2_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)} - \frac{\nabla_\theta\rho_\theta(\tau)\,\nabla_\theta\rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\right)\Big|_{\theta=\widetilde{\theta}}$$

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\tilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln\frac{\rho_{\tilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln\rho_{\tilde{\theta}}(\tau) - \ln\rho_\theta(\tau)\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\tilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\nabla^2_\theta \ln\rho_\theta(\tau)\right]\Big|_{\theta=\tilde{\theta}}$$

$$= -\sum_\tau \rho_{\tilde{\theta}}(\tau)\left(\frac{\nabla^2_\theta\rho_\theta(\tau)}{\rho_\theta(\tau)} - \frac{\nabla_\theta\rho_\theta(\tau)\,\nabla_\theta\rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\right)\Big|_{\theta=\tilde{\theta}}$$

$$= \sum_\tau \rho_{\tilde{\theta}}(\tau)\frac{\nabla_\theta\rho_\theta(\tau)\,\nabla_\theta\rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\Big|_{\theta=\tilde{\theta}}$$

Why?

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\tilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln\frac{\rho_{\tilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln\rho_{\tilde{\theta}}(\tau) - \ln\rho_\theta(\tau)\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\tilde{\theta}} = -\,\mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\nabla^2_\theta \ln\rho_\theta(\tau)\right]\Big|_{\theta=\tilde{\theta}}$$

$$= -\sum_\tau \rho_{\tilde{\theta}}(\tau)\left(\frac{\nabla^2_\theta\rho_\theta(\tau)}{\rho_\theta(\tau)} - \frac{\nabla_\theta\rho_\theta(\tau)\,\nabla_\theta\rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\right)\Big|_{\theta=\tilde{\theta}}$$

$$= \sum_\tau \rho_{\tilde{\theta}}(\tau)\frac{\nabla_\theta\rho_\theta(\tau)\,\nabla_\theta\rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\Big|_{\theta=\tilde{\theta}} \qquad = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\nabla_\theta\ln\rho_\theta(\tau)\left(\nabla_\theta\ln\rho_\theta(\tau)\right)^\top\right]\Big|_{\theta=\tilde{\theta}} \in \mathbb{R}^{d\times d}$$

Why?

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\widetilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \frac{\rho_{\widetilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right] = \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\ln \rho_{\widetilde{\theta}}(\tau) - \ln \rho_\theta(\tau)\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\nabla^2_\theta \ln \rho_\theta(\tau)\right]\Big|_{\theta=\widetilde{\theta}}$$

$$= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\left(\frac{\nabla^2_\theta \rho_\theta(\tau)}{\rho_\theta(\tau)} - \frac{\nabla_\theta \rho_\theta(\tau)\, \nabla_\theta \rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\right)\Big|_{\theta=\widetilde{\theta}}$$

$$= \sum_\tau \rho_{\widetilde{\theta}}(\tau)\frac{\nabla_\theta \rho_\theta(\tau)\, \nabla_\theta \rho_\theta(\tau)^\top}{\left(\rho_\theta(\tau)\right)^2}\Big|_{\theta=\widetilde{\theta}}$$

Why?

$$= \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\nabla_\theta \ln \rho_\theta(\tau)\left(\nabla_\theta \ln \rho_\theta(\tau)\right)^\top\right]\Big|_{\theta=\widetilde{\theta}} \in \mathbb{R}^{d \times d}$$

It's called the Fisher Information Matrix!

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

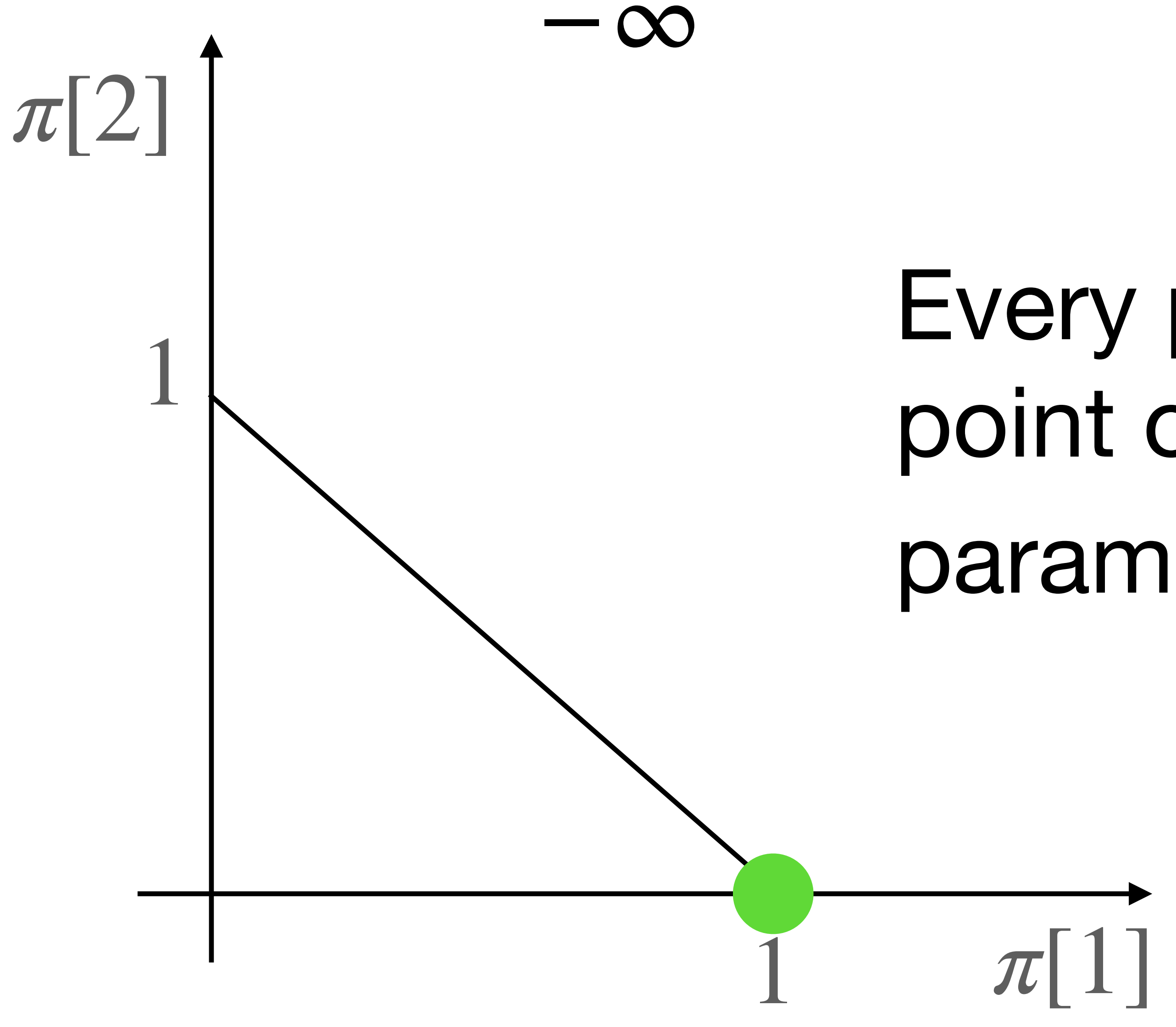
$$(\pi_\theta[1], \pi_\theta[2]) := \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

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

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

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

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

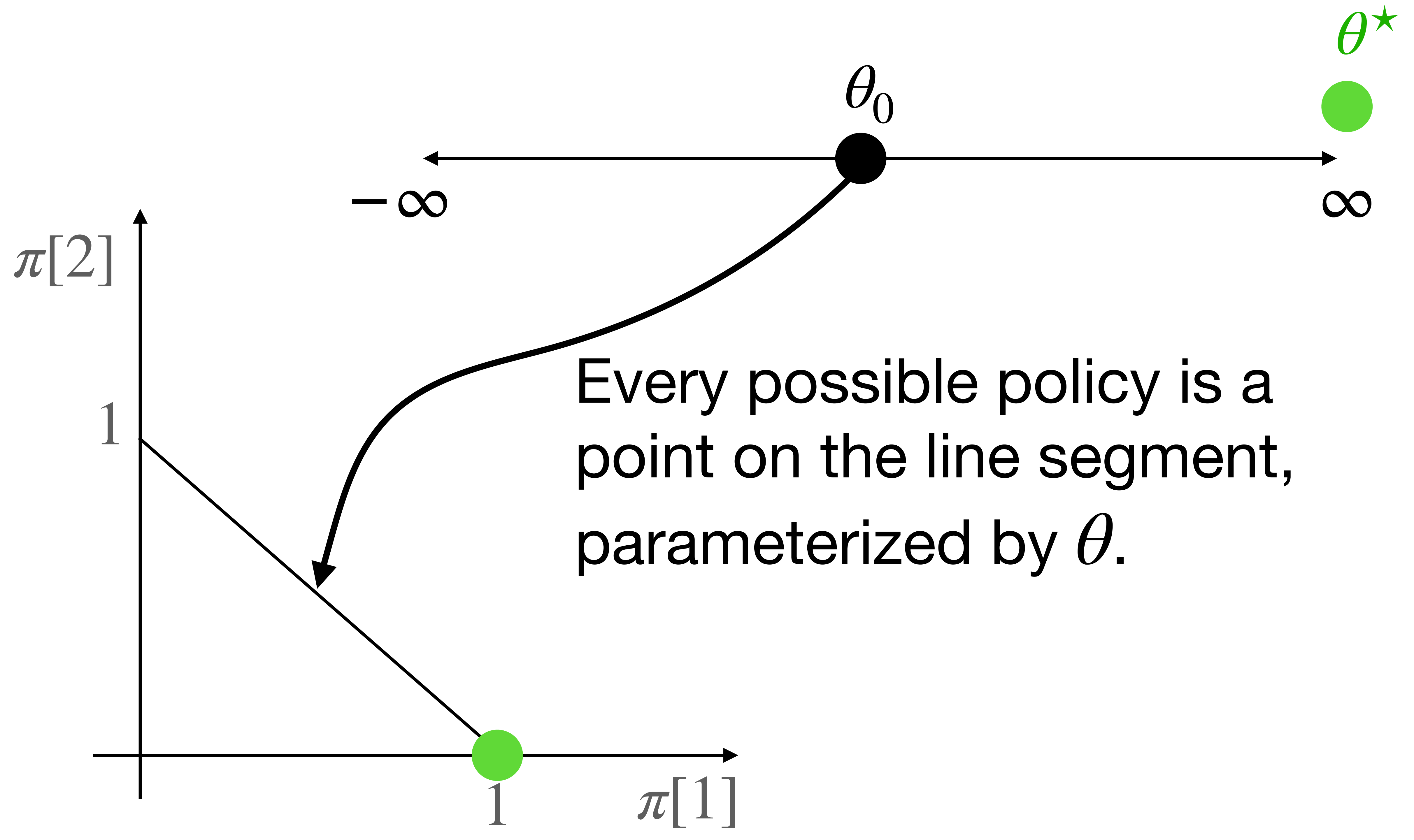


Every possible policy is a point on the line segment, parameterized by θ.

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

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

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

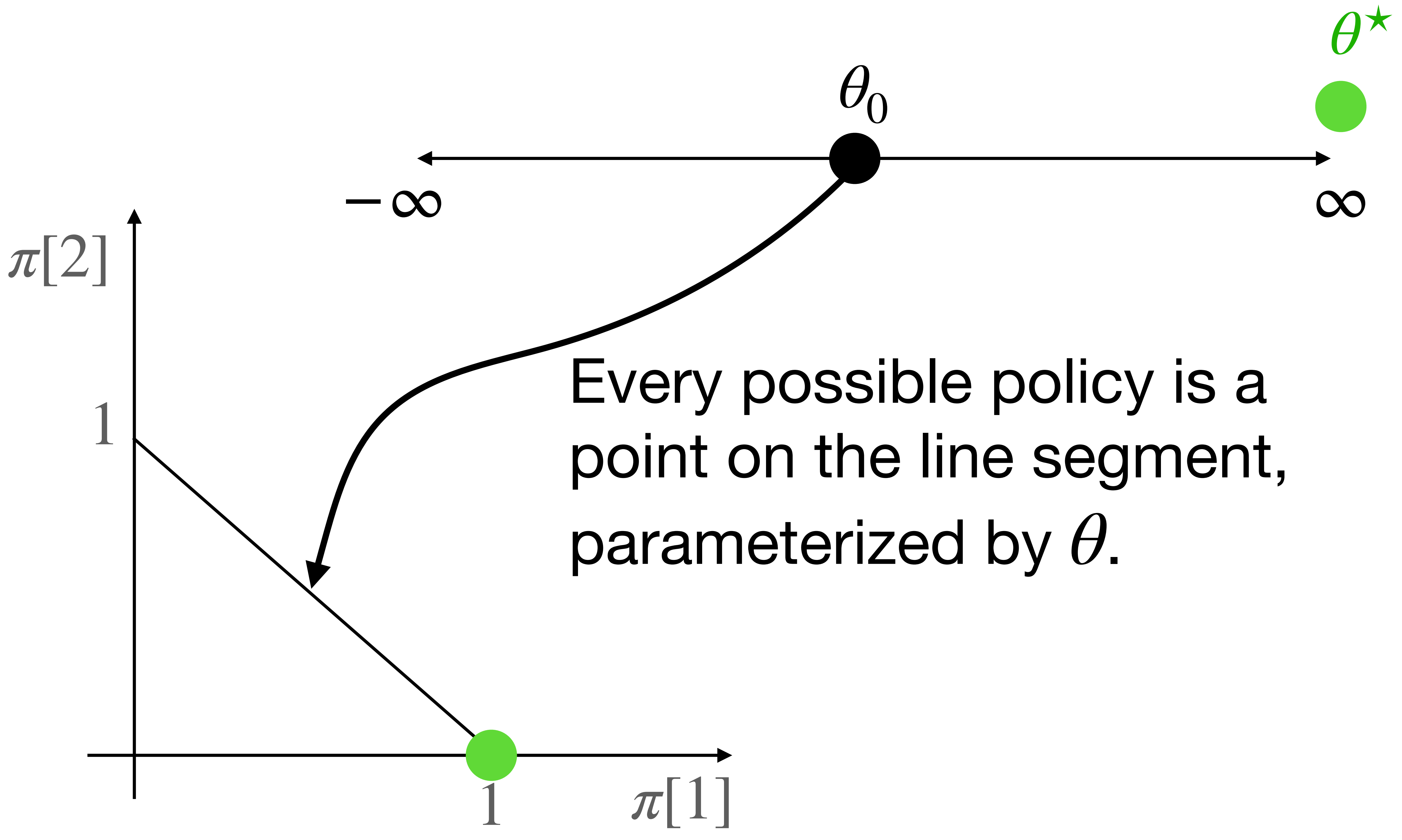


Example of Natural Gradient on 1-d problem: 2 actions, 1 state

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

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

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

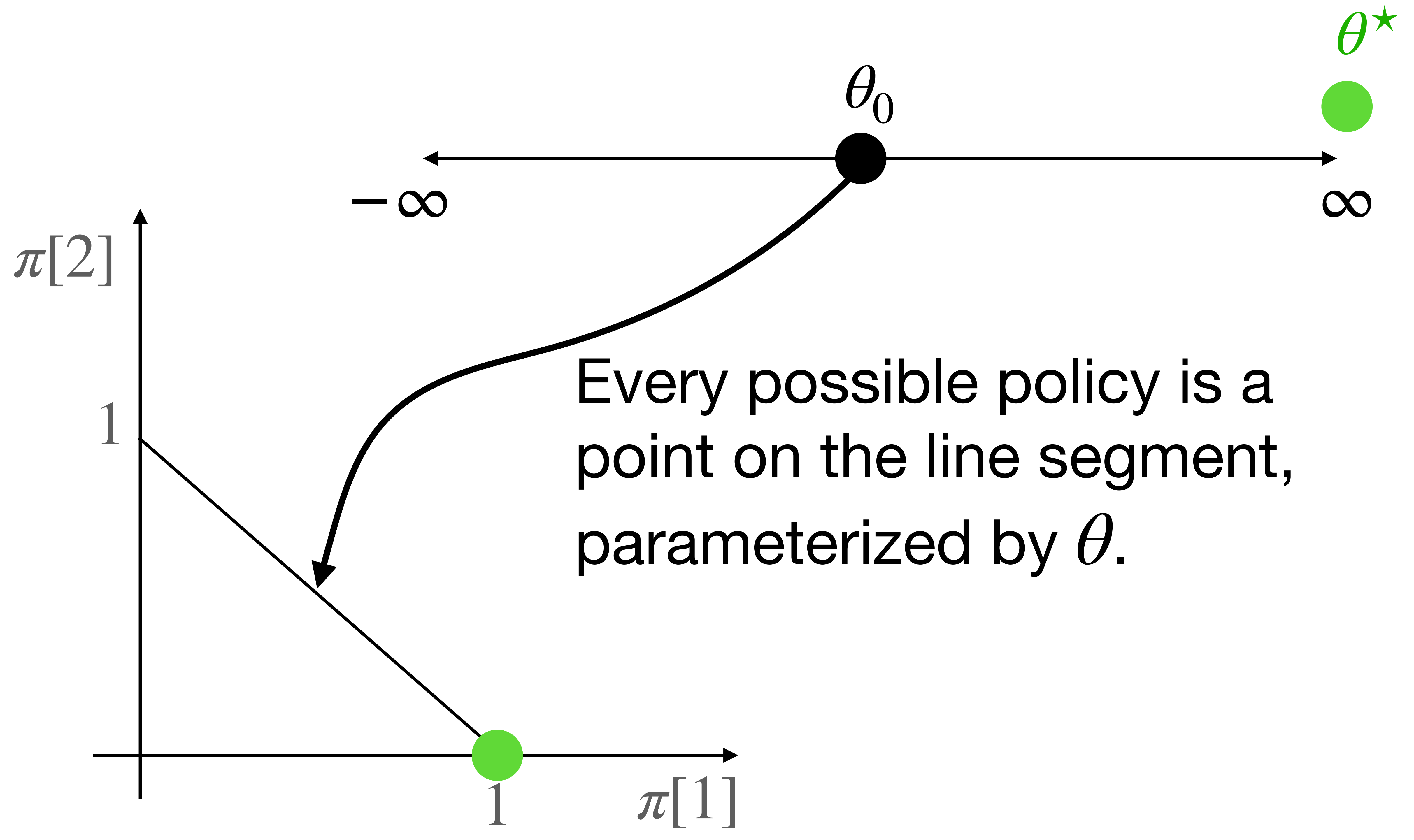


Every possible policy is a point on the line segment, parameterized by θ.

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

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

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



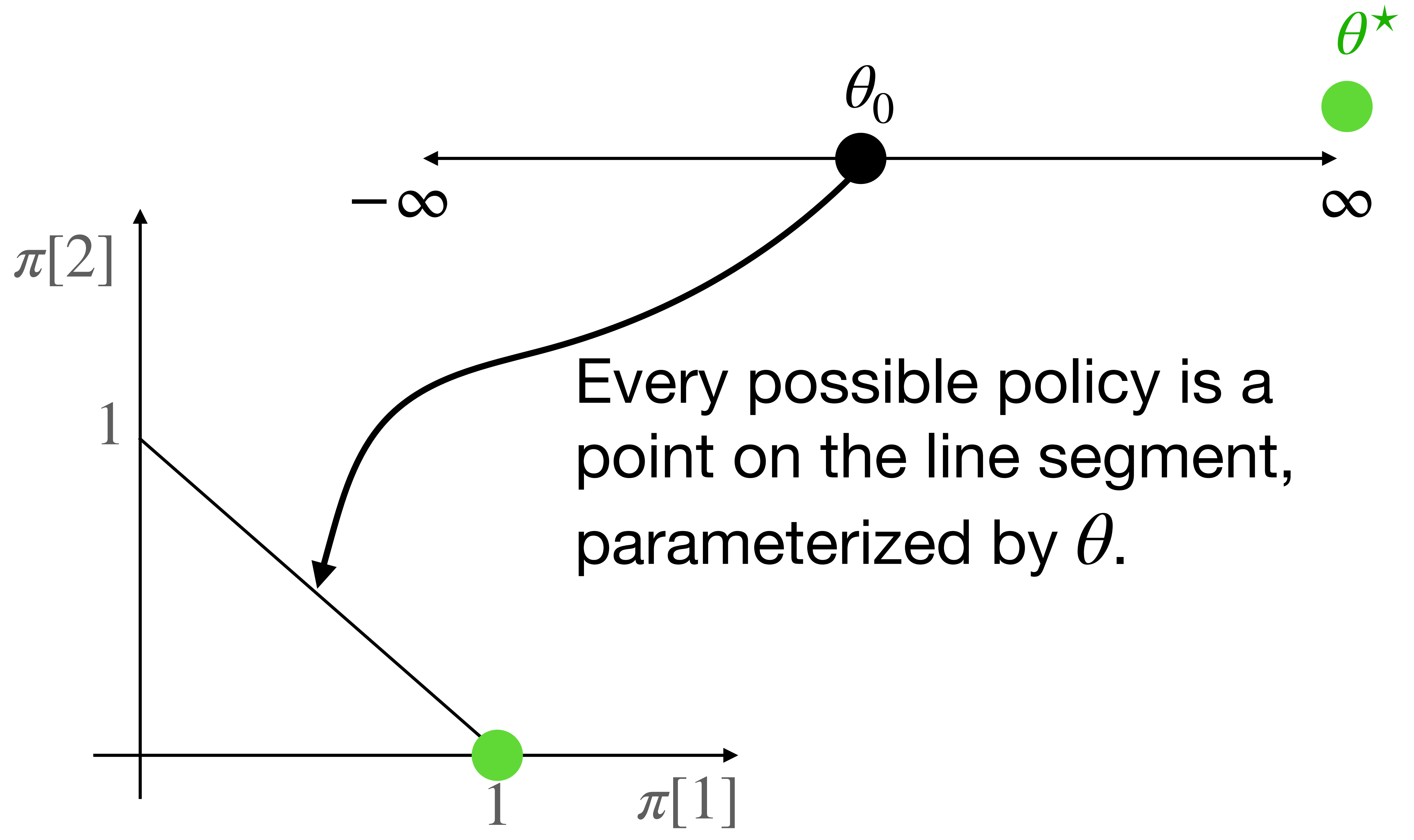
Gradient: $\nabla_\theta J(\theta) = \dfrac{99\exp(\theta)}{(1+\exp(\theta))^2}$

Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99\exp(\theta^k)}{(1+\exp(\theta^k))^2}$

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

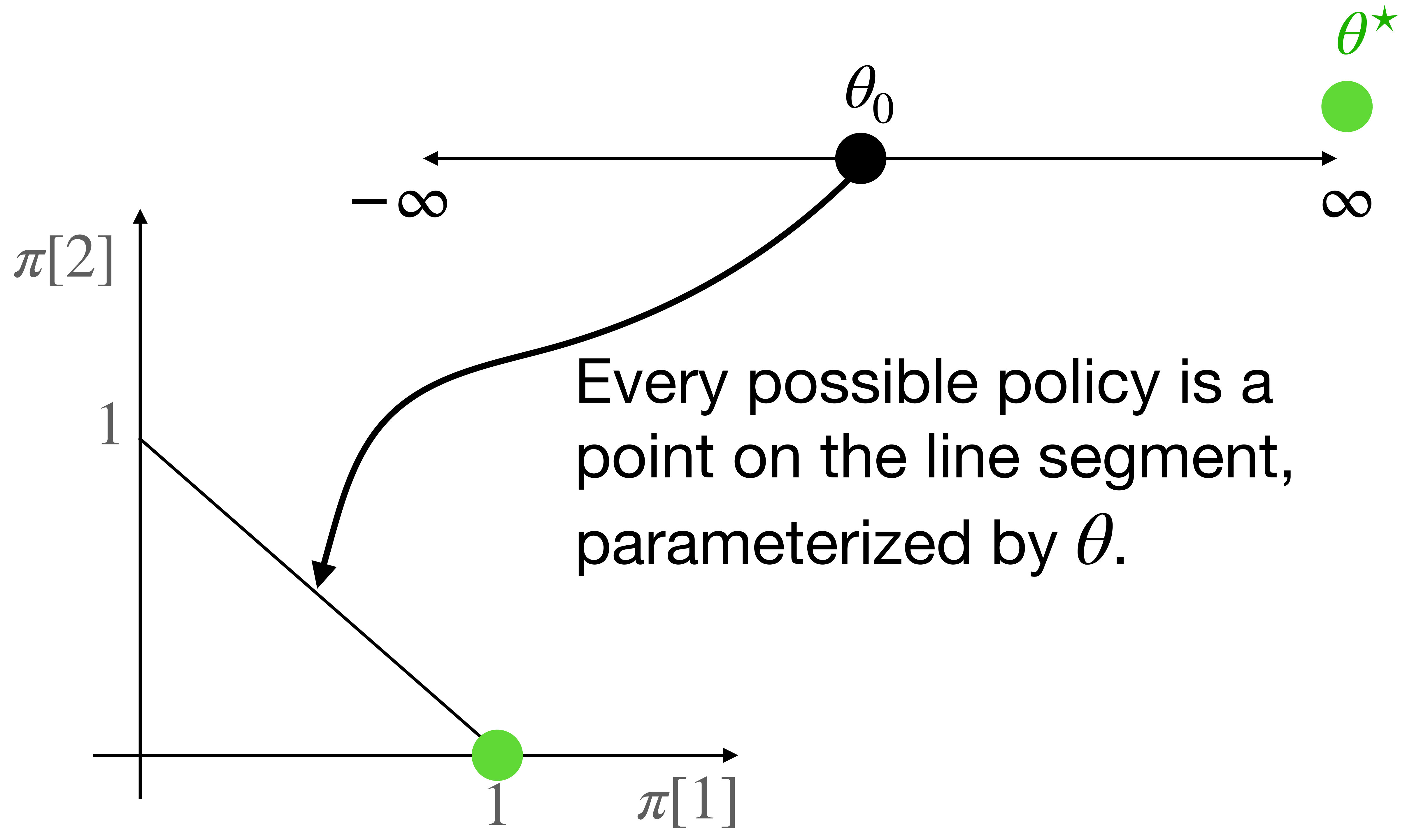
Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99\exp(\theta^k)}{(1+\exp(\theta^k))^2}$

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

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

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

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



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

Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99\exp(\theta^k)}{(1+\exp(\theta^k))^2}$

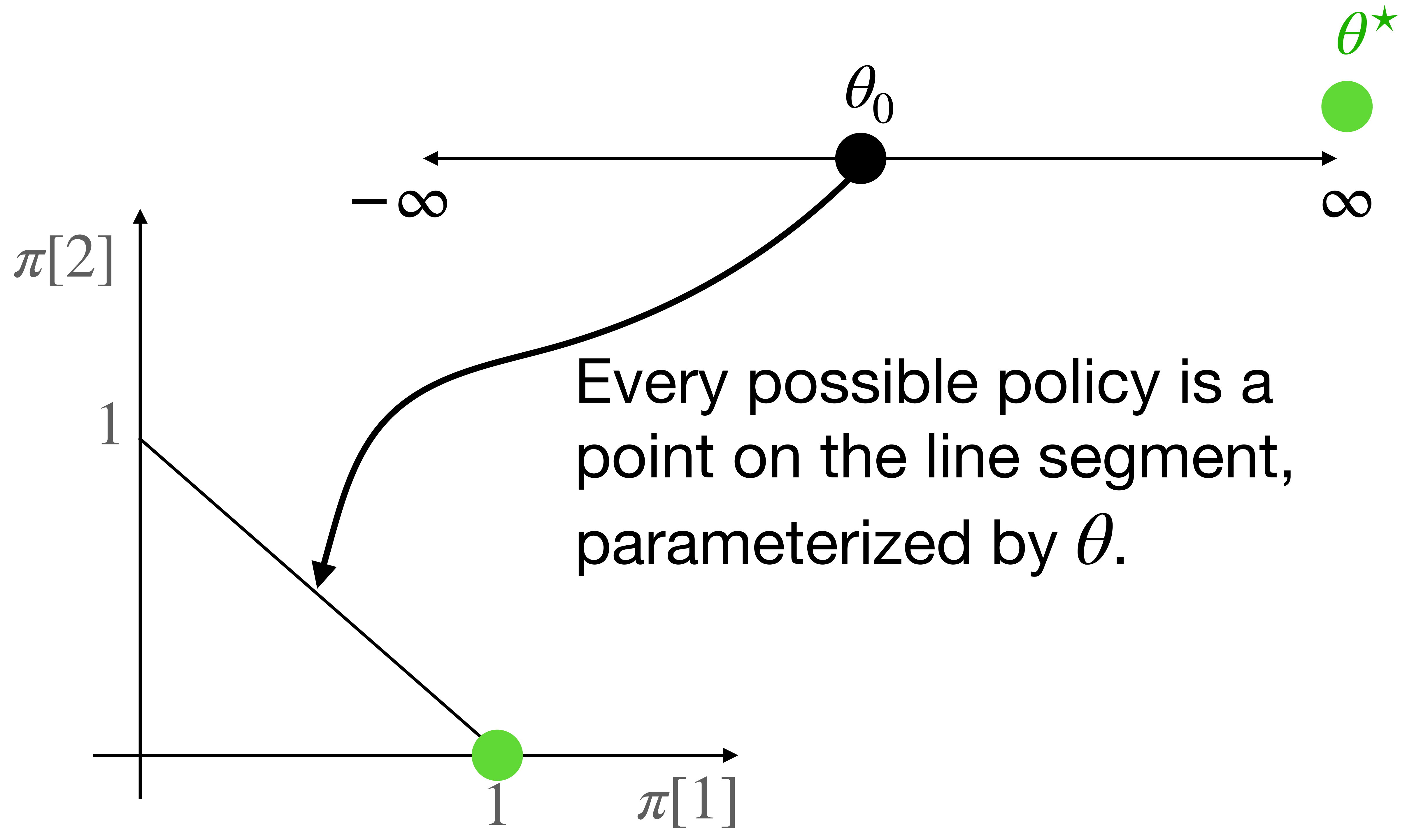
i.e., vanilla GA moves to $\theta = \infty$ with smaller and smaller steps, since $\nabla_\theta J(\theta) \to 0$ as $\theta \to \infty$

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

Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99 \exp(\theta^k)}{(1 + \exp(\theta^k))^2}$

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

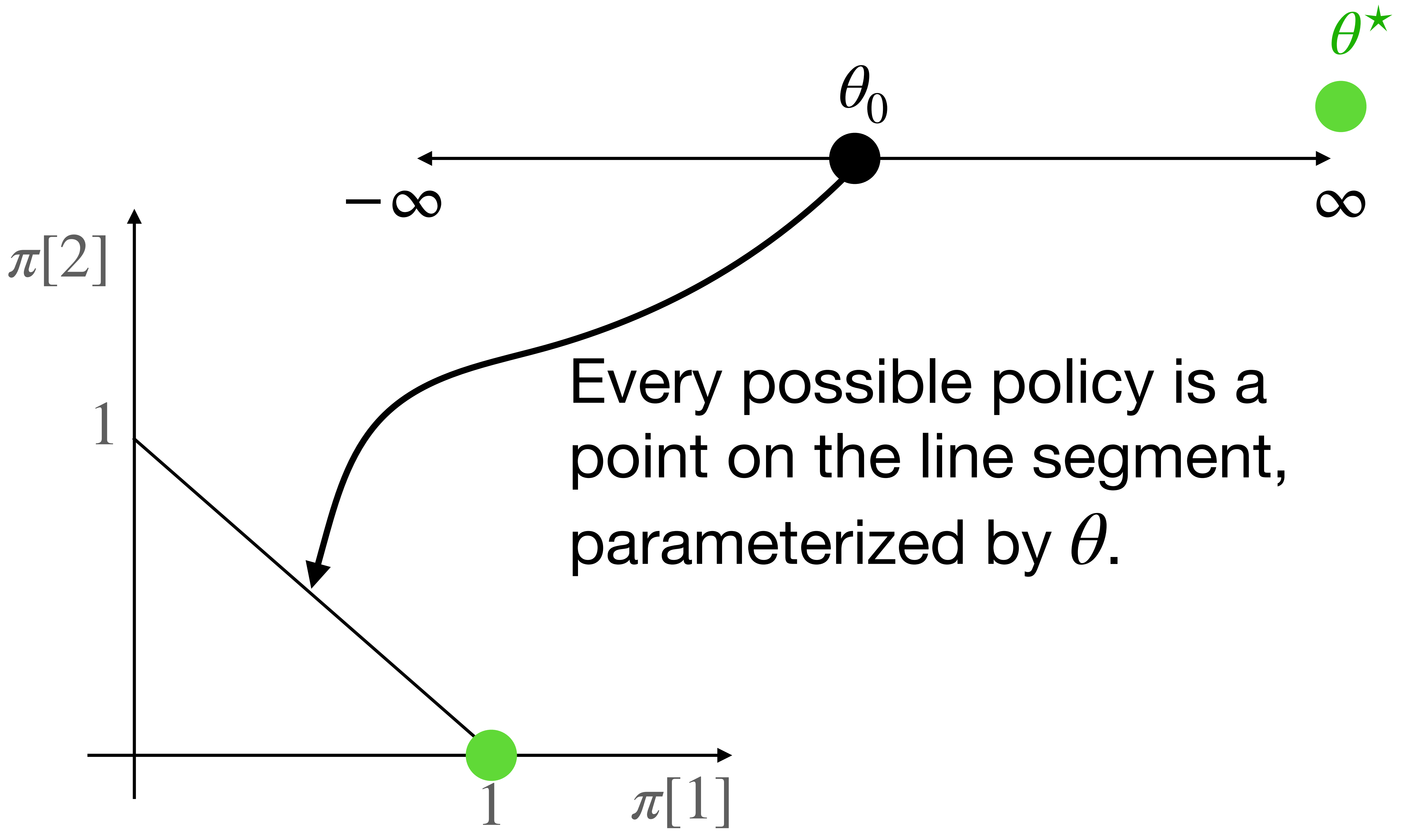
Fisher information scalar: $F_\theta = \dfrac{\exp(\theta)}{(1 + \exp(\theta))^2}$

NPG: $\theta^{k+1} = \theta^k + \eta \dfrac{\nabla_\theta J(\theta^k)}{F_{\theta^k}}$

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

Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99\exp(\theta^k)}{(1+\exp(\theta^k))^2}$

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

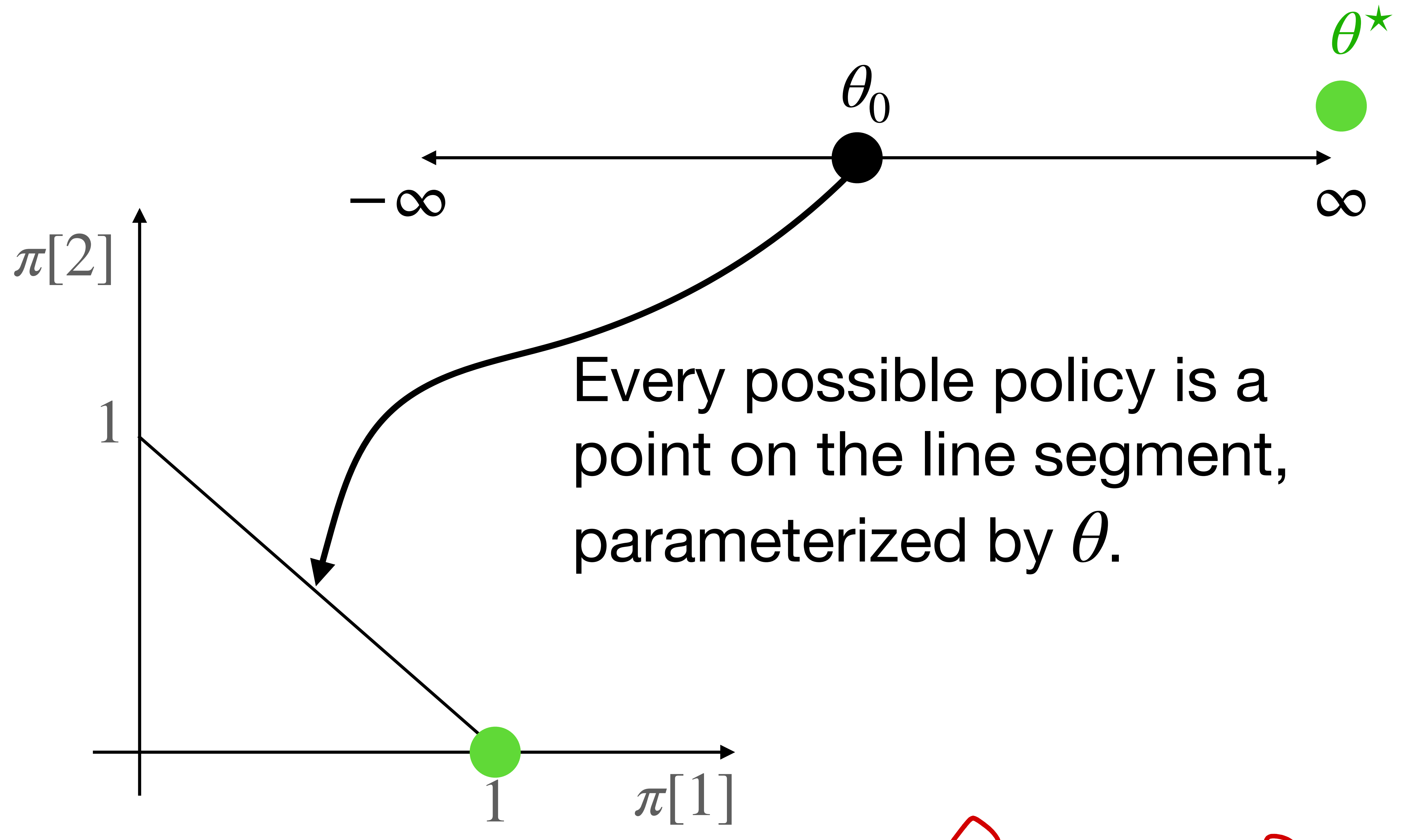
Fisher information scalar: $F_\theta = \dfrac{\exp(\theta)}{(1+\exp(\theta))^2}$

NPG: $\theta^{k+1} = \theta^k + \eta \dfrac{\nabla_\theta J(\theta^k)}{F_{\theta^k}} = \theta_t + \eta \cdot 99$

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.

$P_{\pi_{\theta^{k+1}}} \approx P_{\pi_{\theta^k}}$

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

Exact PG: $\theta^{k+1} = \theta^k + \eta \dfrac{99\exp(\theta^k)}{(1+\exp(\theta^k))^2}$

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

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

NPG: $\theta^{k+1} = \theta^k + \eta \dfrac{\nabla_\theta J(\theta^k)}{F_{\theta^k}} = \theta_t + \eta \cdot 99$

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

Today

- Feedback from last lecture
- Recap
- TRPO -> NPG derivation
- Proximal Policy Optimization (PPO)
- Importance sampling

Back to TRPO/NPG

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   try to approximately solve:
$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$
$$\text{s.t. } KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \le \delta$$
3. Return π_{θ^K}

Back to TRPO/NPG

1. Initialize θ^0
2. For $k = 0, \ldots, K$:

   try to approximately solve:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$

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

3. Return π_{θ^K}

- The difficulty with TRPO and NPG is that they could be computationally costly. Need to solve constrained optimization or matrix inversion ("second order") problems.

Back to TRPO/NPG

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   try to approximately solve:
   $$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$
   $$\text{s.t. } KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) \leq \delta$$
3. Return π_{θ^K}

- The difficulty with TRPO and NPG is that they could be computationally costly. Need to solve constrained optimization or matrix inversion ("second order") problems.
- Can we use a method which only uses gradients?

Back to TRPO/NPG

1. Initialize θ^0
2. For $k = 0,\ldots,K$:

   try to approximately solve:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right]$$

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

3. Return π_{θ^K}

- The difficulty with TRPO and NPG is that they could be computationally costly. Need to solve constrained optimization or matrix inversion ("second order") problems.
- Can we use a method which only uses gradients?

Let's try to use a "Lagrangian relaxation" of TRPO

Proximal Policy Optimization (PPO)

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   try to approximately solve:
   $$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_{\theta}(\cdot|s_h)} \left[A^{\pi_{\theta^k}}(s_h, a_h, h) \right] \right] \underbrace{-\lambda\, KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right)}_{\text{regularization}}$$
3. Return π_{θ^K}

The regularization term is:

$$KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right]$$

The regularization term is:

$$KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right]$$

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 | s_0)P(s_1 | s_0, a_0)\ldots P(s_{H-1} | s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} | s_{H-1})$$

The regularization term is:

$$KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right]$$

$$= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1} \ln \frac{\pi_{\theta^k}(a_h | s_h)}{\pi_\theta(a_h | s_h)}\right]$$

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 | s_0)P(s_1 | s_0, a_0)\ldots P(s_{H-1} | s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} | s_{H-1})$$

The regularization term is:

$$
\begin{aligned}
KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) &= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right] \\
&= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1} \ln \frac{\pi_{\theta^k}(a_h | s_h)}{\pi_\theta(a_h | s_h)}\right] \\
&= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1} \ln \frac{1}{\pi_\theta(a_h | s_h)}\right] + \left[\text{term not a function of } \theta\right]
\end{aligned}
$$

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 | s_0)P(s_1 | s_0, a_0)\ldots P(s_{H-1} | s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} | s_{H-1})$$

Proximal Policy Optimization (PPO)

1. Initialize θ^0
2. For $k = 0, \ldots, K$:
   use SGD to approximately solve:
   $\theta^{k+1} = \arg\max_{\theta} \ell^k(\theta)$
   where:
   $$\ell^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h, a_h, h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h|s_h)}\right]$$
3. Return π_{θ^K}

How do we estimate this objective?

Today

- Feedback from last lecture
- Recap
- TRPO -> NPG derivation
- Proximal Policy Optimization (PPO)
- Importance sampling

SGD and Importance Sampling

SGD and Importance Sampling

- Recall that SGD requires an unbiased estimate of the objective function's gradient

SGD and Importance Sampling

- Recall that SGD requires an unbiased estimate of the objective function's gradient
- This was easy when the objective function was an expectation, and the only θ-dependence appears inside the expectation
  - This was true for supervised learning / ERM
  - Not true for RL, and was part of why we needed likelihood ratio method in REINFORCE

SGD and Importance Sampling

- Recall that SGD requires an unbiased estimate of the objective function's gradient
- This was easy when the objective function was an expectation, and the only θ-dependence appears inside the expectation
  - This was true for supervised learning / ERM
  - Not true for RL, and was part of why we needed likelihood ratio method in REINFORCE
- When not true (as in PPO), we want to make it so, if possible

SGD and Importance Sampling

- Recall that SGD requires an unbiased estimate of the objective function's gradient
- This was easy when the objective function was an expectation, and the only θ-dependence appears inside the expectation
  - This was true for supervised learning / ERM
  - Not true for RL, and was part of why we needed likelihood ratio method in REINFORCE
- When not true (as in PPO), we want to make it so, if possible
- Enter: importance sampling
  - rewrites expectations by changing the distribution the expectation is over
  - we will use this to move that distribution's θ-dependence inside the expectation

SGD and Importance Sampling

- Recall that SGD requires an unbiased estimate of the objective function's gradient
- This was easy when the objective function was an expectation, and the only θ-dependence appears inside the expectation
  - This was true for supervised learning / ERM
  - Not true for RL, and was part of why we needed likelihood ratio method in REINFORCE
- When not true (as in PPO), we want to make it so, if possible
- Enter: importance sampling
  - rewrites expectations by changing the distribution the expectation is over
  - we will use this to move that distribution's θ-dependence inside the expectation
- **Key point**: once all θ-dependence inside objective's expectation,
  - Can estimate objective unbiasedly via sample average
  - Can estimate objective's gradient unbiasedly via gradient of sample average

Importance Sampling

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \widetilde{p}}[f(x)]$.

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \tilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
  - f and $\tilde{p}$ are known.
  - we are not able to collect values of $f(x)$ for $x \sim \tilde{p}$.
    (e.g. we have already collected our data from some costly experiment).

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \tilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
  - f and $\tilde{p}$ are known.
  - we are not able to collect values of $f(x)$ for $x \sim \tilde{p}$.
    (e.g. we have already collected our data from some costly experiment).

- Note: $\mathbb{E}_{x \sim \tilde{p}}\left[f(x)\right] =$

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x\sim\widetilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
    - f and $\widetilde{p}$ are known.
    - we are not able to collect values of $f(x)$ for $x \sim \widetilde{p}$.
      (e.g. we have already collected our data from some costly experiment).
- Note: $\mathbb{E}_{x\sim\widetilde{p}}\left[f(x)\right] = \mathbb{E}_{x\sim p}\left[\frac{\widetilde{p}(x)}{p(x)} f(x)\right]$

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \widetilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
  - f and $\widetilde{p}$ are known.
  - we are not able to collect values of $f(x)$ for $x \sim \widetilde{p}$.
    (e.g. we have already collected our data from some costly experiment).
- Note: $\mathbb{E}_{x \sim \widetilde{p}}\left[f(x)\right] = \mathbb{E}_{x \sim p}\left[\frac{\widetilde{p}(x)}{p(x)} f(x)\right]$
- So an unbiased estimate of $\mathbb{E}_{x \sim \widetilde{p}}[f(x)]$ is given by $\frac{1}{N}\sum_{i=1}^{N} \frac{\widetilde{p}(x_i)}{p(x_i)} f(x_i)$

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \widetilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
  - f and $\widetilde{p}$ are known.
  - we are not able to collect values of $f(x)$ for $x \sim \widetilde{p}$.
    (e.g. we have already collected our data from some costly experiment).
- Note: $\mathbb{E}_{x \sim \widetilde{p}}\left[f(x)\right] = \mathbb{E}_{x \sim p}\left[\frac{\widetilde{p}(x)}{p(x)} f(x)\right]$
- So an unbiased estimate of $\mathbb{E}_{x \sim \widetilde{p}}[f(x)]$ is given by $\frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{p}(x_i)}{p(x_i)} f(x_i)$
- Terminology:
  - $\widetilde{p}(x)$ is the target distribution
  - $p(x)$ is the proposal distribution
  - $\widetilde{p}(x)/p(x)$ is the likelihood ratio or importance weight

Importance Sampling

- Suppose we seek to estimate $\mathbb{E}_{x \sim \tilde{p}}[f(x)]$.
- Assume: we have an (i.i.d.) dataset $x_1, \ldots x_N$, where $x_i \sim p$, where p is known, and
  - f and $\tilde{p}$ are known.
  - we are not able to collect values of $f(x)$ for $x \sim \tilde{p}$.
    (e.g. we have already collected our data from some costly experiment).
- Note: $\mathbb{E}_{x \sim \tilde{p}} \left[f(x) \right] = \mathbb{E}_{x \sim p} \left[\frac{\tilde{p}(x)}{p(x)} f(x) \right]$
- So an unbiased estimate of $\mathbb{E}_{x \sim \tilde{p}}[f(x)]$ is given by $\frac{1}{N} \sum_{i=1}^{N} \frac{\tilde{p}(x_i)}{p(x_i)} f(x_i)$
- Terminology:
  - $\tilde{p}(x)$ is the target distribution
  - $p(x)$ is the proposal distribution
  - $\tilde{p}(x)/p(x)$ is the likelihood ratio or importance weight
- What about the variance of this estimator?

Importance Sampling & Variance

Back to Estimating $\ell^k(\theta)$

Back to Estimating $\ell^k(\theta)$

- To estimate

$$\ell^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h\,|\,s_h)}\right]$$

Back to Estimating $\ell^k(\theta)$

- To estimate

$$\ell^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h\,|\,s_h)}\right]$$

- we will use importance sampling:

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\frac{\pi_\theta(a_h\,|\,s_h)}{\pi_{\theta^k}(a_h\,|\,s_h)}A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h\,|\,s_h)}\right]$$

Back to Estimating $\ell^k(\theta)$

- To estimate

$$\ell^k(\theta) := \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h|s_h)}\right]$$

- we will use importance sampling:

$$= \mathbb{E}_{s_0,\ldots,s_{H-1}\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta^k}(\cdot|s_h)}\left[\frac{\pi_\theta(a_h|s_h)}{\pi_{\theta^k}(a_h|s_h)}A^{\pi_{\theta^k}}(s_h,a_h,h)\right]\right] - \lambda\mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\ln\frac{1}{\pi_\theta(a_h|s_h)}\right]$$

$$= \mathbb{E}_{\tau\sim\rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1}\left(\frac{\pi_\theta(a_h|s_h)}{\pi_{\theta^k}(a_h|s_h)}A^{\pi_{\theta^k}}(s_h,a_h,h) - \lambda\ln\frac{1}{\pi_\theta(a_h|s_h)}\right)\right]$$

Estimating $\ell^k(\theta)$ and its gradient

Estimating $\ell^k(\theta)$ and its gradient

1. Using N trajectories sampled under $\rho_{\pi_{\theta^k}}$ to learn a $\widetilde{b}_h$

   $\widetilde{b}(s,h) \approx V_h^{\pi_{\theta^k}}(s)$

Estimating $\ell^k(\theta)$ and its gradient

1. Using N trajectories sampled under $\rho_{\pi_{\theta^k}}$ to learn a $\widetilde{b}_h$

   $\widetilde{b}(s,h) \approx V_h^{\pi_{\theta^k}}(s)$

2. Obtain M NEW trajectories $\tau_1, \ldots \tau_M \sim \rho_{\pi_{\theta^k}}$

   Set $\widehat{\ell}^{\,k}(\theta) = \frac{1}{M}\sum_{m=1}^{M}\sum_{h=0}^{H-1}\left(\frac{\pi_\theta(a_h^m \mid s_h^m)}{\pi_{\theta^k}(a_h^m \mid s_h^m)}\left(R_h(\tau_m) - \widetilde{b}(s_h^m, h)\right) - \lambda \ln \frac{1}{\pi_\theta(a_h^m \mid s_h^m)}\right)$

   for SGD, use gradient: $g(\theta) := \nabla_\theta \widehat{\ell}^{\,k}(\theta)$

Estimating $\ell^k(\theta)$ and its gradient

1. Using N trajectories sampled under $\rho_{\pi_{\theta^k}}$ to learn a $\widetilde{b}_h$

   $\widetilde{b}(s, h) \approx V_h^{\pi_{\theta^k}}(s)$

2. Obtain M NEW trajectories $\tau_1, \ldots \tau_M \sim \rho_{\pi_{\theta^k}}$

   Set $\widehat{\ell}^k(\theta) = \frac{1}{M} \sum_{m=1}^{M} \sum_{h=0}^{H-1} \left(\frac{\pi_\theta(a_h^m \mid s_h^m)}{\pi_{\theta^k}(a_h^m \mid s_h^m)} \left(R_h(\tau_m) - \widetilde{b}(s_h^m, h) \right) - \lambda \ln \frac{1}{\pi_\theta(a_h^m \mid s_h^m)} \right)$

   for SGD, use gradient: $g(\theta) := \nabla_\theta \widehat{\ell}^k(\theta)$

$g(\theta^k)$ is unbiased for $\nabla_\theta \ell^k(\theta)\Big|_{\theta=\theta^k}$

Summary:

1. NPG: a simpler way to do TRPO, a “pre-conditioned” gradient method.
2. PPO: “first order” approximation to TRPO

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy