Trust Region Policy Optimization
&
the Natural Policy Gradient

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning

Fall 2022

Today

- Announcements:
  If you are an undergraduate student at Harvard and are possibly interested in pursuing research, formally or informally, with the ML foundations group, please fill in the following form: https://forms.gle/yCiTfbXn31x2RQtHA

- Recap

- Today:
  1. Convergence of Fitted Policy Iteration
  2. Trust Region Policy Optimization

Recap

[Policy Eval Subroutine]: TD Learning for "tabular" case

[Iterative Policy Eval Subroutine/TD]

input: policy π, sample size N, init w_0

1. Sample trajectories $\tau_1, \ldots \tau_N \sim \rho_\pi$ which gives us a dataset D

   (each trajectory is of the form $\tau_i = \{s_0, a_0, r_0, \ldots s_{H-1}, a_{H-1}, r_{H-1},\}$)
2. For k = $0,\ldots,K$:
   1. Sample a transition $(s_h, r_h, s_{h+1},) \in D$ and update:

$$V_{k+1}(s_h) = V_k(s_h) - \eta_k \Big(V_k(s_h) - \big(r_h + V_k(s_{h+1})\big) \Big)$$

3. Return the function V_K as an estimate of V^π

Q-Learning for "tabular" case

[Iterative Policy Eval Subroutine/TD]

input: **offline dataset** $\tau_1, \ldots \tau_N \sim \rho_{\pi_{data}}$

1. For k = $0,\ldots,K$:
   1. Sample a transition $(s_h, a_h, r_h, s_{h+1}) \in D$ and update:

$$Q_{k+1}(s_h, a_h) = Q_k(s_h, a_h) - \eta_k\Big(Q_k(s_h, a_h) - \big(r_h + \max_{a'} Q_k(s_{h+1}, a')\big)\Big)$$

2. Return the function Q_K as an estimate of $Q^\star$

Fitted Policy Iteration:
(aka Approximate Policy Iteration API)

1. Initialize staring policy π_0, samples size M
2. For k = 0, … :
    1. [Q-Evaluation Subroutine]

       Using M sampled trajectories, $\tau_1, \ldots \tau_N \sim \rho_{\pi_k}$,

       $\widetilde{Q}_k(s,a) \approx Q_h^{\pi_k}(s,a)$
    2. Policy Update

       $\pi_{k+1}(s) := \arg\max_a \widetilde{Q}^{\pi_k}(s,a)$
3. Return $\widetilde{Q}_K$ and π_K as an estimate of $Q^\star$ and $\pi^\star$

Recap on Supervised Learning: regression

We have a data distribution $\mathscr{D}$, $x_i \sim \mathscr{D}, y_i = f^\star(x_i) + \epsilon_i$, where noise $\mathbb{E}[\epsilon_i] = 0, |\epsilon_i| \leq c$

$$\hat{f} = \arg\min_{f \in \mathscr{F}} \sum_{i=1}^{N} \left(f(x_i) - y_i\right)^2$$

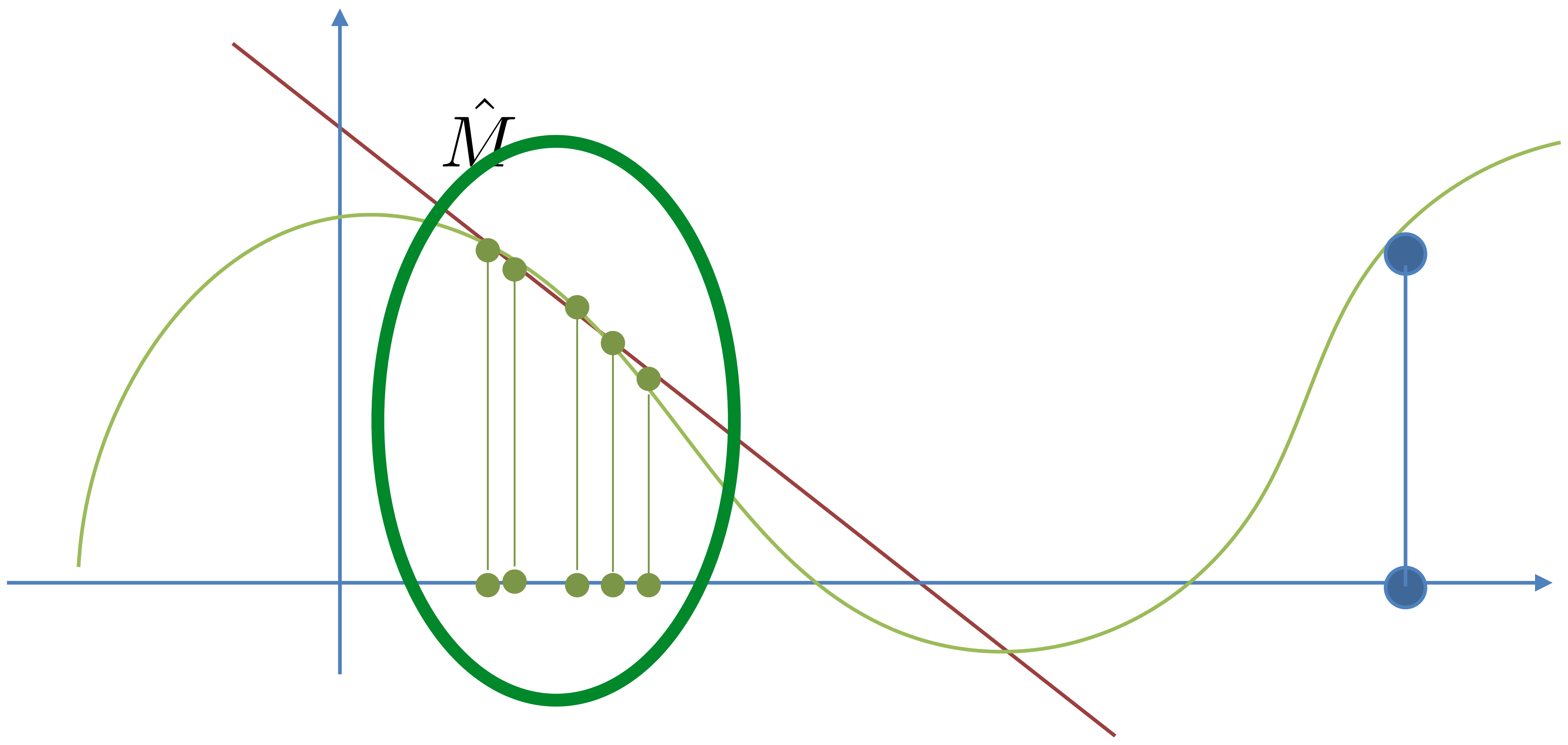
Supervised learning theory (e.g., VC theory) says that we can indeed **generalize**, i.e., we can predict well **under the same distribution:**

Assume $f^\star \in \mathscr{F}$ (this is called realizability), we can expect:

$$\mathbb{E}_{x \sim \mathscr{D}} \left(\hat{f}(x) - f^\star(x)\right)^2 \leq \delta$$

Supervise Learning can fail if there is train-test distribution mismatch

However, for some $\mathscr{D}' \neq \mathscr{D}$, $\mathbb{E}_{x \sim \mathscr{D}'} (f(x) - f^\star(x))^2$ might be arbitrarily large



Deeper neural nets and larger datasets are typically not enough to address "distribution shift"

Fitted Policy Improvement Guarantees

- For all k, suppose that:

$$E_{\tau\sim\rho_{\pi_k}}\left[\sum_{h=1}^{H}\left(\widetilde{Q}_k(s_h,a_h)-Q_h^{\pi_k}(s_h,a_h)\right)^2\right]\leq\delta,\text{ and }\ \max_{s,a}|\widetilde{Q}_k(s_h,a_h)-Q_h^{\pi_k}(s_h,a_h)|\leq\delta_\infty$$

- 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:] We have that:

- One step performance degradation is bounded by the worst case error:
  $$Q^{k+1}(s)\geq Q^k(s)-2H\delta_\infty \text{ (and equality possible in some examples).}$$
- For large enough K, final performance also governed by the worst case error:
  $$Q^{\pi_K}(s)\geq Q^\star(s)-2H^2\delta_\infty$$
- (Intuition) If it somehow turns out that, for all iterations k, the density under the next policy, uniformly does not differ from that of previous policy, i.e. that
  $$\max_{s,a,h}\left(\frac{\Pr(s_h=s,a_h=a\,|\,\pi_{k+1})}{\Pr(s_h=s,a_h=a\,|\,\pi_k)}\right)\leq C_\infty$$
  then we can bound our sub-optimality by the average case error:
  $$Q^{\pi_K}(s)\geq Q^\star(s)-2H^2\cdot C_\infty\cdot\delta$$

Today:

Optimality in Markov Decision Processes

Outline:

1. Quick intro on KL-divergence
   & the visitation measure
2. A Trust-Region Formulation for Policy Optimization
3. Algorithm: Natural Policy Gradient

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 $KL(P|Q) = \|\mu_1 - \mu_2\|_2^2/\sigma^2$

Fact:

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

Outline:

1. Quick intro on KL-divergence
2. A Trust-Region Formulation for Policy Optimization
3. Algorithm: Natural Policy Gradient

A trust region formulation for policy update:

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

$$\max_{\theta} \mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\theta_t}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a) \right]$$

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

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)

Some Helpful Notation: Visitation Measures

- 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:

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

$$= E\left[\sum_{h=0}^{H-1} r(s_h, a_h) \middle| \mu_0, \pi_\theta\right]$$

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

Visitation Measures: the discounted case

- 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:
$$d_\mu^\pi(s,a) = (1-\gamma)\sum_{h=0}^{\infty} \gamma^h \mathbb{P}_h(s,a \mid \mu,\pi)$$
- With this def, we have:
$$J(\theta) := E_{s_0 \sim \mu_0}\left[V^{\pi_\theta}(s_0)\right]$$
$$= E\left[\sum_{h=0}^{\infty} \gamma^h r_h \middle| \mu_0, \pi_\theta\right]$$
$$= \frac{1}{1-\gamma} \cdot \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} E_{a \sim \pi(s)}\left[r(s,a)\right]$$

Equivalently,

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…

Outline:

1. Quick intro on KL-divergence
2. A Trust-Region Formulation for Policy Optimization
3. Algorithm: Natural Policy Gradient

A trust region formulation for policy update:

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

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

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

High-level strategy:

1. First-order Taylor expansion on the objective at θ_t

2.second-order Taylor expansion of the constraint at θ_t

Simplify Objective Function

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

Since the objective is also non-linear,
let's do first order-talyor expansion on it:

$$H\ \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right] \approx H\ \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(s)} A^{\pi_{\theta_t}}(s,a)\right] + \underbrace{\mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(s)} \nabla_\theta \ln \pi_{\theta_t}(a\,|\,s) A^{\pi_{\theta_t}}(s,a)\right]}_{\nabla_\theta J(\pi_{\theta_t})} \cdot (\theta - \theta_t)$$

$$= \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$

Simplify Constraint via second-order Taylor Expansion:

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

$$\ell(\theta) \approx \ell(\theta_t) + \nabla \ell(\theta_t)^\top (\theta - \theta_t) + \frac{1}{2}(\theta - \theta_t)^\top \nabla_\theta^2 \ell(\theta_t)(\theta - \theta_t)$$

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

We will show that $\nabla_\theta \ell(\theta_t) = 0$, and $\nabla^2 \ell(\theta_t)$ has a nice form!

The gradient of the KL-divergence is zero at θ_t

Change from trajectory distribution to state-action distribution:

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

$$= H \mathbb{E}_{s_h, a_h \sim d_\mu^{\pi_{\theta_t}}} \left[\ln \frac{\pi_{\theta_t}(a_h | s_h)}{\pi_\theta(a_h | s_h)} \right] := \ell(\theta)$$

$$\nabla_\theta \ell(\theta) |_{\theta=\theta_t} = H \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \sum_a \pi_{\theta_t}(a | s) \left(- \nabla_\theta \ln \pi_\theta(a_h | s_h) |_{\theta=\theta_t} \right)$$

$$= - H \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \sum_a \pi_{\theta_t}(a | s) \frac{\nabla_\theta \pi_{\theta_t}(a | s)}{\pi_{\theta_t}(a | s)} = 0$$

Let's compute the Hessian of the KL-divergence at θ_t

$$H \cdot \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}} \left[\ln \frac{\pi_{\theta_t}(a_h \mid s_h)}{\pi_\theta(a_h \mid s_h)} \right] := \ell(\theta)$$

$$\frac{1}{H} \nabla_\theta^2 \ell(\theta) \mid_{\theta=\theta_t} = \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \sum_a \pi_{\theta_t}(a \mid s) \left(-\nabla_\theta^2 \ln \pi_\theta(a \mid s) \mid_{\theta=\theta_t} \right)$$

$$= -\mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \sum_a \pi_{\theta_t}(a \mid s) \left(\frac{\nabla_\theta^2 \pi_{\theta_t}(a \mid s)}{\pi_{\theta_t}(a \mid s)} - \frac{\nabla_\theta \pi_{\theta_t}(a \mid s) \nabla_\theta \pi_{\theta_t}(a \mid s)^\top}{\pi_{\theta_t}^2(a \mid s)} \right)$$

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

It's called fisher Information Matrix!

Summary so far:

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:

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

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

Outlines

1. Quick intro on KL-divergence

2. A Trust-Region Formulation for Policy Optimization

3. Algorithm: Natural Policy Gradient

Put everything together, we get:

(dropping the H factors) At iteration t, we update to θ_{t+1} via:

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

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

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

Indeed this gives us:

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$$

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

Algorithm: Natural Policy Gradient

Initialize θ_0

For t = 0, …

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

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

Natural Gradient Ascent: $\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$

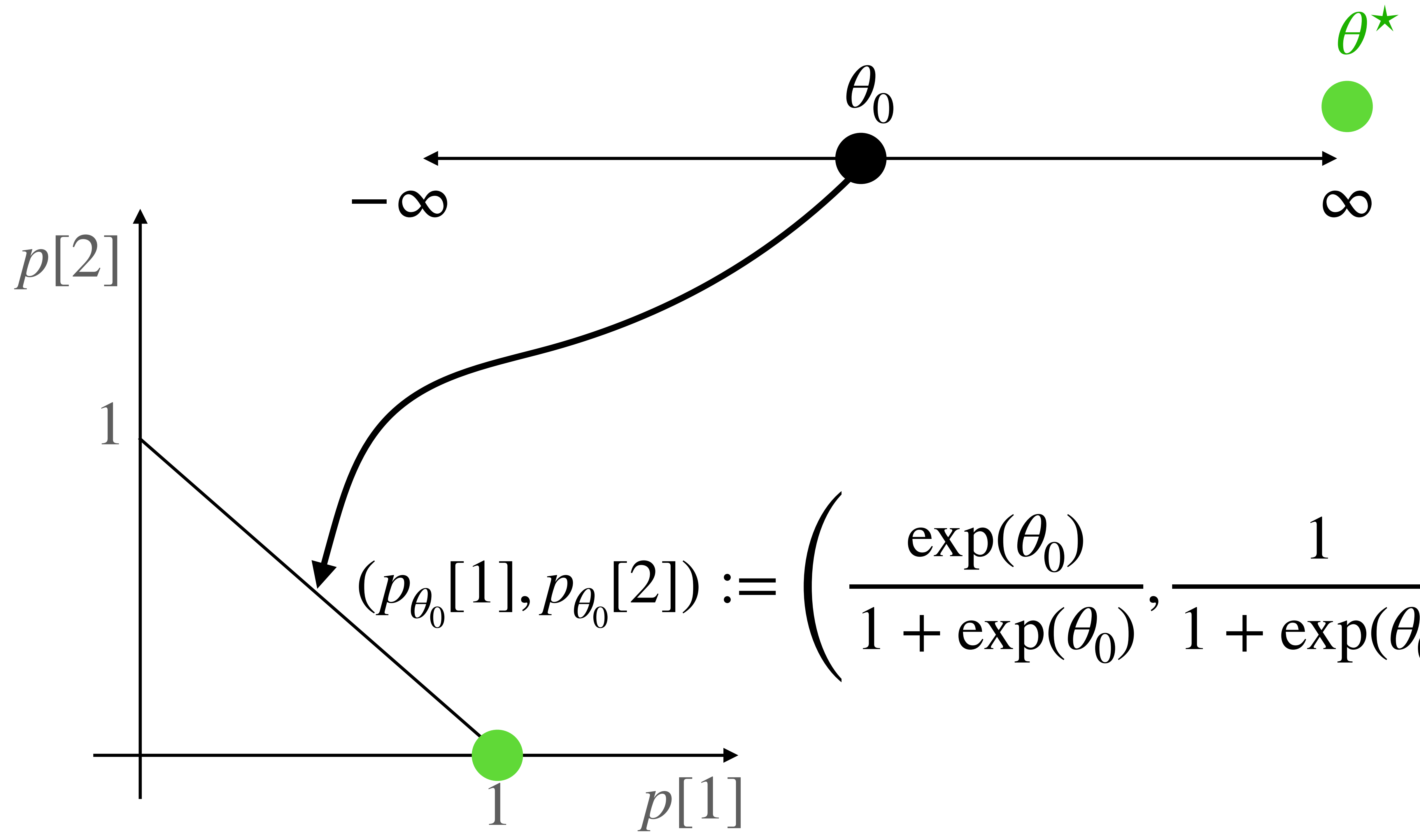
Where $\eta = \sqrt{\dfrac{\delta}{\nabla_\theta J(\pi_{\theta_t})^\top F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})}}$

(We will implement it in HW4 on Cartpole)

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

Hence: $f_{\theta_0} \to 0^+$, as $\theta_0 \to \infty$

NPG: $\theta_1 = \theta_0 + \eta \dfrac{g'(\theta_0)}{f_{\theta_0}}$

GA: $\theta_1 = \theta_0 + \eta g'(\theta_0)$

i.e., Plain GA in θ will move to $\theta = \infty$ at a constant speed,
while Natural GA can traverse faster and faster when θ gets bigger
(subject to the same learning rate)

Summary for NPG:

Trust Region Policy Optimization and NPG

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right] \longrightarrow$$ First-order Taylor expansion at θ_t

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

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

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

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

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t}) \longleftarrow$$

NPG

(Exercise: work out the $\arg\max_\theta$)

An extension of NPG (even faster in practice):

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

fifth attempt (new): **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 $\arg\max_\theta \ell(\theta)$

Next a few lectures:

Imitation Learning
(Learning from Demonstrations)

Can we learn a good policy purely from expert demonstrations?

Summary:

1. Convergence of Fitted Policy Iteration
2. Trust Region Policy Optimization
   1. Quick intro on KL-divergence
   2. TRPO formulation

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