CS 277: Control and Reinforcement Learning

Winter 2022

Lecture 6: Advanced Model-Free Methods

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine

Logistics

quizzes

- Quiz 3 due Friday

assignments

- Assignment 2 to be published soon

Today's lecture

Convergence

Continuous action spaces

Trust-region methods

Backup operator

- Value recursion: $V_\pi(s) = \mathbb{E}_{(a|s)\sim\pi}[r(s,a) + \gamma \mathbb{E}_{(s'|s,a)\sim p}[V_\pi(s')]] = \mathcal{T}_\pi[V_\pi](s)$

  linear backup operator

- In matrix notation:

$$\vec{v}_\pi = \vec{r}_\pi + \gamma P_\pi \vec{v}_\pi$$

$$\boxed{\underset{|S|}{\vec{r}}} + \gamma \boxed{\underset{|S|\times|S|}{P}} \cdot \boxed{\underset{|S|}{\vec{v}}}$$

  - where $r_\pi(s) = \mathbb{E}_{(a|s)\sim\pi}[r(s,a)]$, $P_\pi(s,s') = \mathbb{E}_{(a|s)\sim\pi}[p(s'|s,a)]$

$$\begin{array}{cc} & s' \\ s & \boxed{\underset{|S|\times|S|}{P}} \end{array}$$

- Can be solved with linear algebra:

$$\vec{v}_\pi = (I - \gamma P_\pi)^{-1} \vec{r}_\pi$$

  largest eigenvalue magnitude

- The inverse always exists because P_π has spectral radius 1

Bellman operator

- Bellman operator: $\mathcal{T}[V](s) = \max_a r(s,a) + \gamma \mathbb{E}_{(s'|s,a)\sim p}[V(s')]$
  - Action-value version: $\mathcal{T}[Q](s,a) = r(s,a) + \gamma \mathbb{E}_{(s'|s,a)\sim p}[\max_{a'} Q(s',a')]$
- Value Iteration = iteratively apply $\mathcal{T}$
- Why is this guaranteed to converge? $\mathcal{T}$ is a contraction:

$$\|\mathcal{T}[V_1] - \mathcal{T}[V_2]\|_\infty = \max_{s,a} \gamma \mathbb{E}_{(s'|s,a)\sim p}[V_1(s') - V_2(s')] \leq \gamma \|V_1(s') - V_2(s')\|_\infty$$

replace $\mathbb{E}$ with $\max_{s'}$

- $V^* = \mathcal{T}[V^*]$ is the unique fixed point

MF

θ

DP

π'

max

Q-Learning convergence

- Q-Learning: $Q(s,a) \to_\alpha r + \gamma \max_{a'} Q(s',a')$

  in expectation, this is $\mathcal{T}[Q]$

  - In iteration i, use learning rate α_i

- Robbins–Monro: converges to Q^* with probability 1 (almost surely) if:
  - $\sum_i \alpha_i^2 < \infty$, implying $\alpha_i \to 0$ faster than $i^{-1/2}$
  - $\sum_i \alpha_i = \infty$, implying $\alpha_i \to 0$ not faster than i^{-1}

- Example: $\alpha_i = i^{-1}$ (like in averaging)

MF

θ

DP

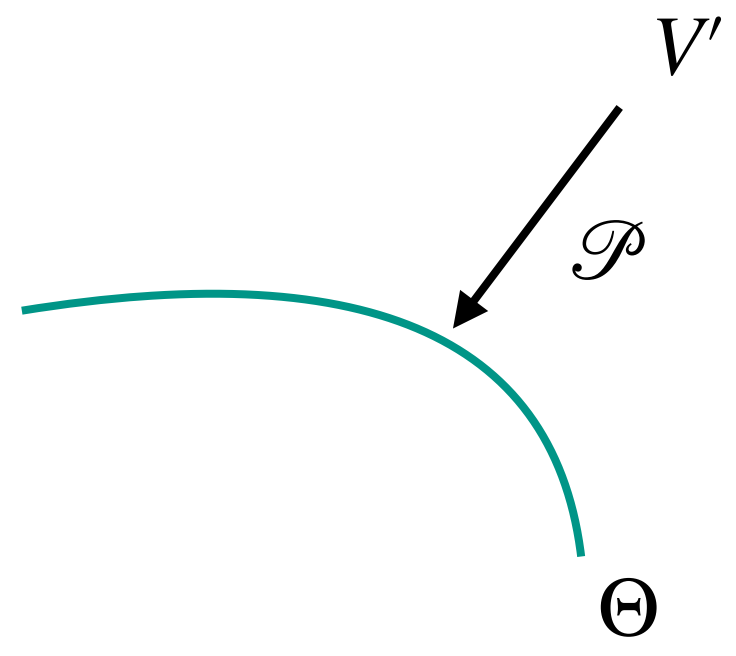
π'

max

Fitted Value Iteration

- Bellman (TD) error: $\mathscr{T}[V_{\bar{\theta}}](s) - V_\theta$

- Minimizing the square error is a projection



$$\mathscr{P}[V'] = \min_{\theta \in \Theta} \|V' - V_\theta\|_2^2$$

- If Θ is convex, the projection is a non-expansion

$$\|\mathscr{P}[V_1'] - \mathscr{P}[V_2']\|_2^2 \leq \|V_1' - V_2'\|_2^2$$

- Composition of contractions contracts; but norms mismatch ($\mathscr{T}$: L_∞; $\mathscr{P}$: L_2)
  - So $\mathscr{P}\mathscr{T}$ is generally not a contraction $\Rightarrow$ no convergence guarantee

But isn't DQN just SGD?

Algorithm DQN

Initialize θ, set $\bar{\theta} \leftarrow \theta$

$s \leftarrow$ reset state

for each interaction step

  Sample $a \sim \epsilon$-greedy for $Q_\theta(s, \cdot)$

  Get reward r and observe next state s'

  Add (s, a, r, s') to replay buffer $\mathcal{D}$

  Sample batch $(\vec{s}, \vec{a}, \vec{r}, \vec{s}') \sim \mathcal{D}$

$$y_i \leftarrow \begin{cases} r_i & s'_i \text{ terminal} \\ r_i + \gamma \max_{a'} Q_{\bar{\theta}}(s'_i, a') & \text{otherwise} \end{cases}$$

  Descend $\mathcal{L}_\theta = (\vec{y} - Q_\theta(\vec{s}, \vec{a}))^2$

  every T_{target} steps, set $\bar{\theta} \leftarrow \theta$

  $s \leftarrow$ reset state if s' terminal, else $s \leftarrow s'$

moving target
≠ SGD

MF
θ
DP
π'
max

Is PG just SGD?

MF

θ

DP

π'

max

Algorithm REINFORCE

Initialize π_θ

repeat

Roll out $\xi \sim p_\theta$

Update with gradient $g \leftarrow R(\xi) \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)$

- The gradient is unbiased for $\nabla_\theta J_\theta$
- The objective J_θ changes with θ, but so does $\mathbb{E}_{x\sim D}[L_\theta(x)]$ in general ML
- But the data distribution changes
  - No convergence guarantees (not even local!)

Today's lecture

Convergence

Continuous action spaces

Trust-region methods

Continuous actions spaces

can do for large / continuous action spaces?

- What do we need for policy-based methods?
  - For rollouts: given s, sample from $\pi_\theta(a \mid s)$ ✓
  - For policy update: given s and a, compute $\nabla_\theta \log \pi_\theta(a \mid s)$ ✓
- What do we need for value-based methods?
  - For rollouts: given s, compute $\arg\max_a Q_\theta(s, a)$ ✗
  - For value updates: given s, compute $\max_a Q_\theta(s, a)$ ✗
- How can we use value-based methods with continuous action spaces?

Idea 1: DQN with stochastic optimization

- If we can't enumerate A, let's sample $a_1, \ldots, a_k$ and take $\max_i Q(s, a_i)$
  - Sample from what distribution?
- Let's find an ad-hoc approximately greedy policy π
  - Run value-based algorithm; whenever it needs $\max_a Q(s, a)$:

Algorithm Stochastic optimization

Initialize π
repeat
  Sample $a_1, \ldots, a_k \sim \pi$
  Select k/c top values $Q(s, a_i)$ for $i = 1, \ldots k$
  Fit π to these "elites"

Idea 2: easily maximizable Q

MF

θ

DP

π'

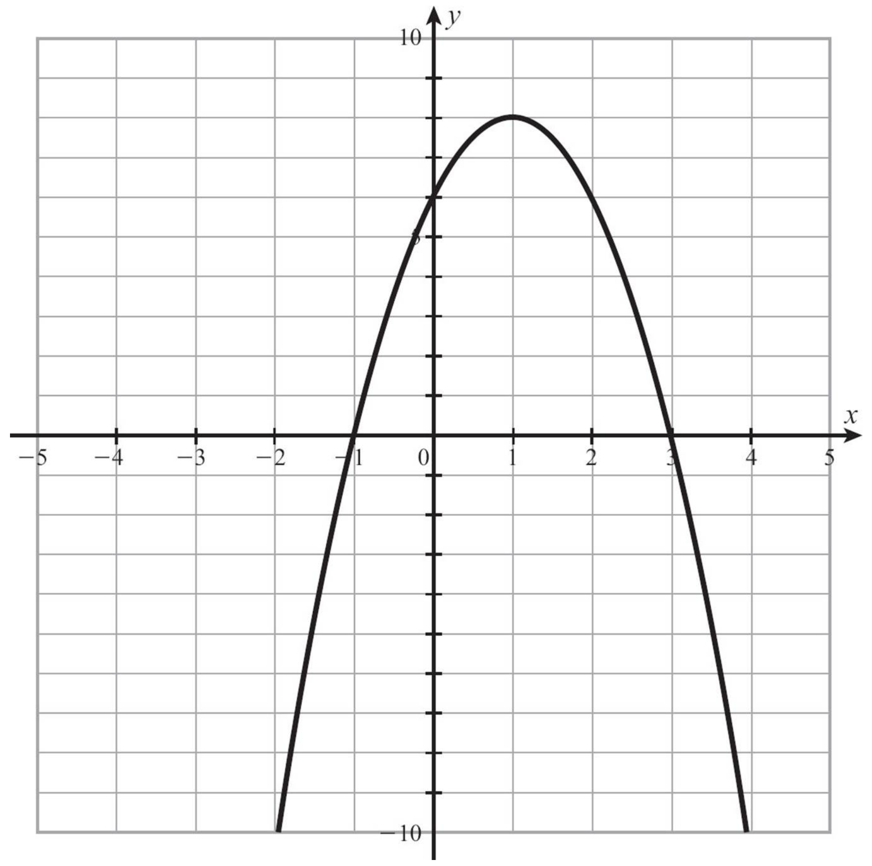
max

- Represent Q_θ in a way that is directly maximizable

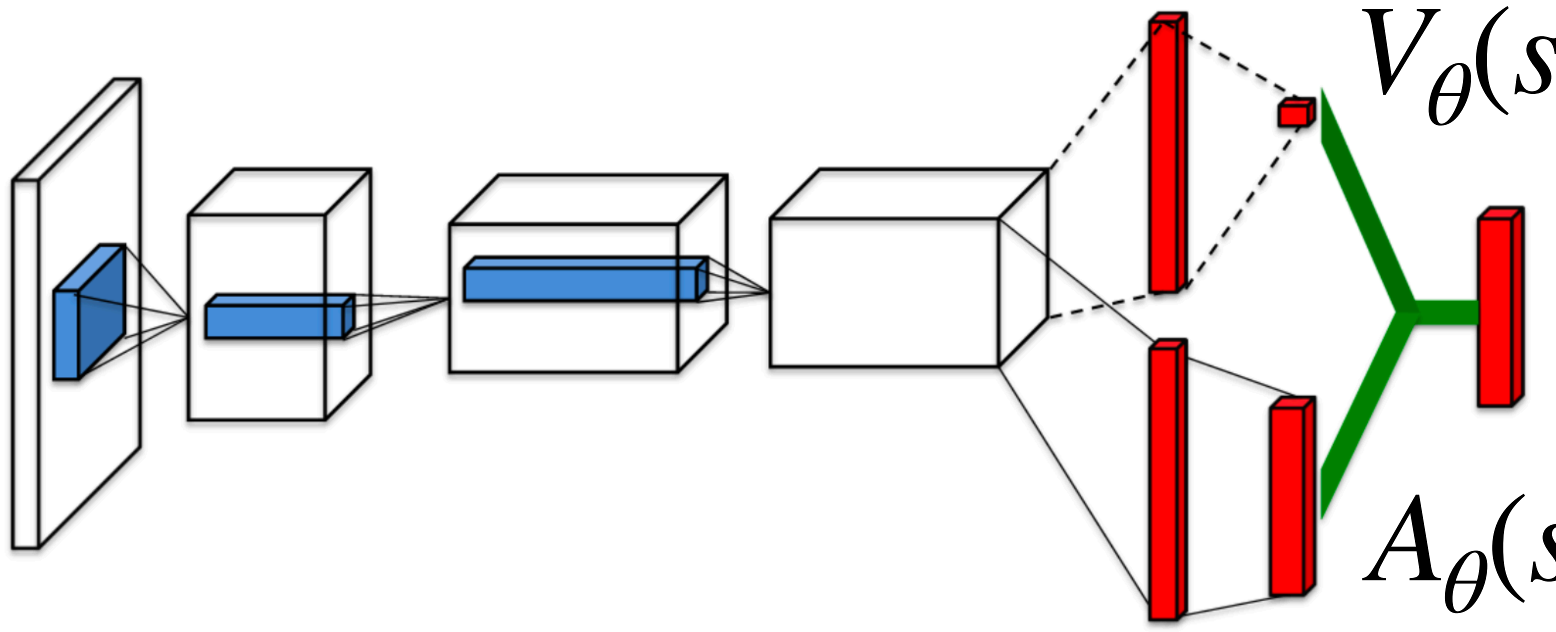
- Example: quadratic $Q_\theta(s,a) = -\frac{1}{2}(a-\mu_\theta(s))^\intercal P_\theta(s)(a-\mu_\theta(s)) + V_\theta(s)$

$$\arg\max_a Q_\theta(s,a) = \mu_\theta(s)$$

$$\max_a Q_\theta(s,a) = V_\theta(s)$$



- Possible architecture: dueling network



$V_\theta(s)$

$A_\theta(s,a) = -\frac{1}{2}(a-\mu_\theta(s))^\intercal P_\theta(s)(a-\mu_\theta(s))$

[Gu et al., 2016]

Idea 3: learn optimizing policy

- Previous methods: represent a Q maximizer or train one ad-hoc
- More general method: let a deterministic $\mu_\theta(s)$ learn to maximize $Q_\phi(s, a)$
  - This makes it an Actor–Critic method
- Deterministic Policy Gradient Theorem:

$$\begin{aligned}\nabla_\theta V_{\mu_\theta}(s) = \nabla_\theta Q_{\mu_\theta}(s, \mu_\theta(s)) &= \nabla_\theta Q_{\mu_{\bar\theta}}(s, \mu_\theta(s)) + \nabla_\theta Q_{\mu_\theta}(s, \mu_{\bar\theta}(s)) \\ &= \nabla_\theta Q_{\mu_{\bar\theta}}(s, \mu_\theta(s)) + \gamma \mathbb{E}_{(s'|s,\mu_\theta(s))\sim p}[\nabla_\theta V_{\mu_\theta}(s')]\end{aligned}$$

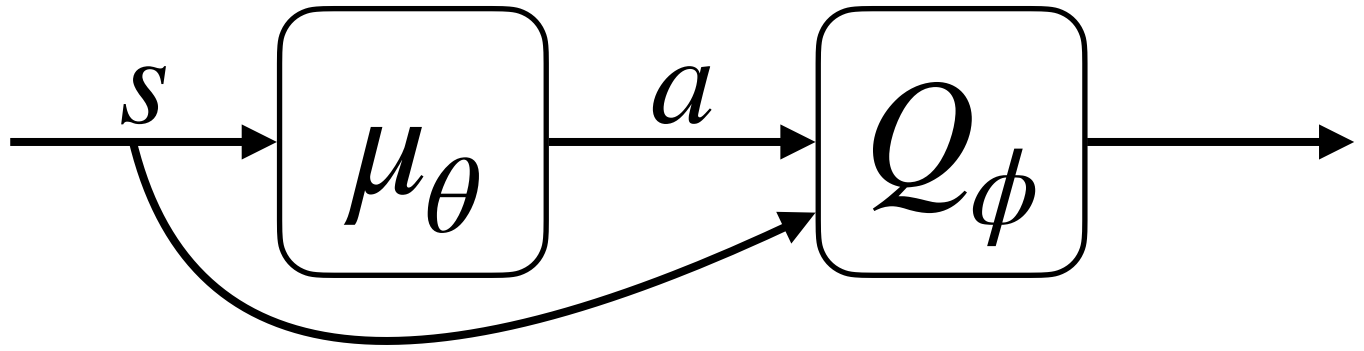
pseudo-reward

$$\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta}[\nabla_\theta Q_{\mu_{\bar\theta}}(s_t, \mu_\theta(s_t))] = \cancel{\frac{1}{1-\gamma}} \mathbb{E}_{s\sim p_\theta}[\nabla_\theta Q_{\mu_{\bar\theta}}(s, \mu_\theta(s))]$$

$t \sim \mathrm{Geo}(1-\gamma)$

[Silver et al., 2014]

Deep Deterministic Policy Gradient (DDPG)



- Evaluating Q: feed actor $\mu_\theta(s)$ into critic $Q_\phi(s, a)$

- Back-propagation (chain rule):

$\longleftarrow \nabla_\theta \mu_\theta(s) \quad \nabla_a Q_\phi(s, a) \longleftarrow$

$$\nabla_\theta J_\theta = \mathbb{E}_{s \sim p_\theta}[\,\nabla_\theta Q_\phi(s, \mu_\theta(s))]$$
$$= \mathbb{E}_{s \sim p_\theta}[\,\nabla_\theta \mu_\theta(s)\,\nabla_a Q_\phi(s, a = \mu_\theta(s))]$$

- DDPG:
  - Train critic Q_ϕ: TD policy evaluation
  - Train actor π_θ: ascend $Q_\phi(s, \mu_\theta)$ with gradient through μ_θ

MF

θ

DP

π'

max

[Lillicrap et al., 2016]

Today's lecture

Convergence

Continuous action spaces

Trust-region methods

Importance Sampling

- Suppose you want to estimate $\mathbb{E}_{x \sim p}[f(x)]$
  - but only have samples $x \sim p'$

- Importance sampling:

$$\mathbb{E}_{x \sim p}[f(x)] = \mathbb{E}_{x \sim p'}\left[\frac{p(x)}{p'(x)} f(x)\right]$$

  - Importance (IS) weights: $\rho(x) = \dfrac{p(x)}{p'(x)}$
  - Estimate: $\rho(x) f(x)$ with $x \sim p'$

IS application 1: multi-step Q-Learning

- n-step Q-Learning: $Q(s_t, a_t) \to \sum_{\Delta t=0}^{n-1} \gamma^{\Delta t} r_{t+\Delta t} + \gamma^n \max_a Q(s_{t+n}, a)$

- On-policy data $(r_t, r_{t+1}, \dots, r_{t+n-1}, s_{t+n})$: a_t can be anything
  - but must have $a_{t+\Delta t} = \arg\max_a Q(s_{t+\Delta t}, a)$ for LHS = E[RHS] (Bellman optimality)

- To be off-policy: update $Q(s_t, a_t) \to \sum_{\Delta t=0}^{n-1} \gamma^{\Delta t} \rho_t^{\Delta t} r_{t+\Delta t} + \gamma^n \max_a Q(s_{t+n}, a)$
  - For data from π', with $\rho_t^{\Delta t} = \prod_{i=t+1}^{t+\Delta t} \frac{\pi(a_i | s_i)}{\pi'(a_i | s_i)}$

MF

θ

DP

π'

max

IS application 2: off-policy policy evaluation

MF
θ
DP
π'
max

- Estimate $J_\pi = \mathbb{E}_{\xi \sim p_\pi}[R(\xi)]$ off-policy: $J_\pi = \mathbb{E}_{\xi \sim p_{\pi'}}[\rho^\pi_{\pi'}(\xi) R(\xi)]$
  - with $\rho^\pi_{\pi'}(\xi) = \frac{p_\pi(\xi)}{p_{\pi'}(\xi)} = \prod_t \frac{\pi(a_t | s_t)}{\pi'(a_t | s_t)}$ $p(s' | s, a)$ **cancels out**

- $\rho(\xi)$ can be very large or small ⇒ high variance

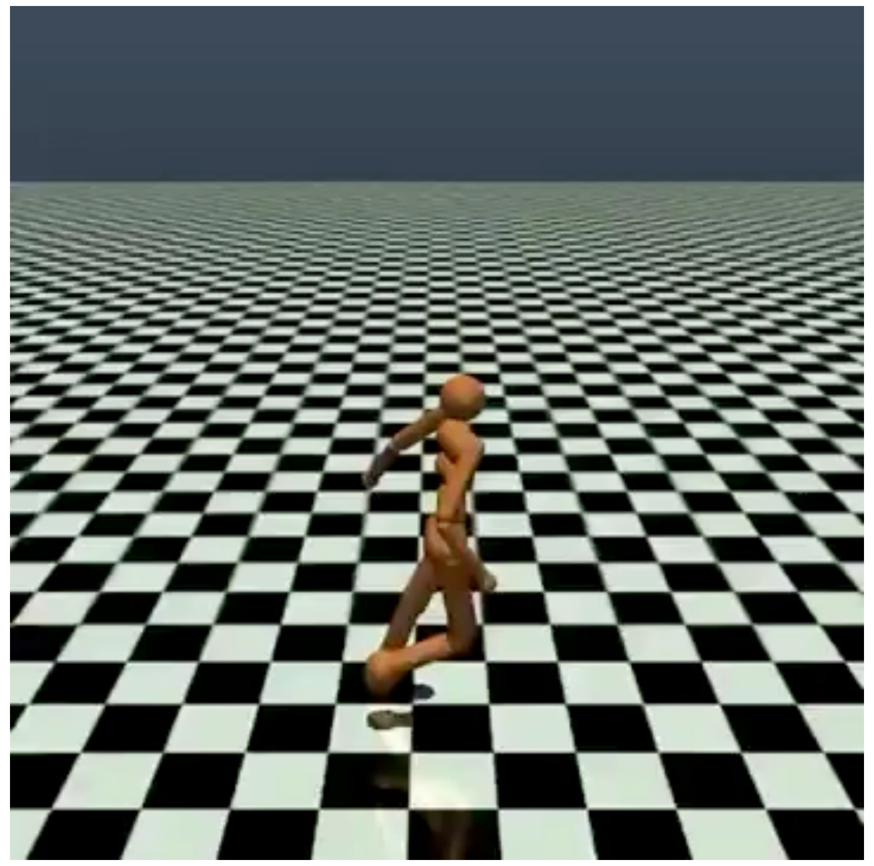
- Some reduction: r_t is not affected by future actions

$$J_\pi = \sum_t \mathbb{E}_{\xi_{\leq t} \sim p_{\pi'}}[\gamma^t \rho^\pi_{\pi'}(\xi_{\leq t}) r_t] = \sum_t \mathbb{E}_{\xi_{\leq t} \sim p_{\pi'}}\left[\gamma^t r_t \prod_{t' \leq t} \frac{\pi(a_{t'} | s_{t'})}{\pi'(a_{t'} | s_{t'})}\right]$$

[Precup et al., 2000]

IS application 3: Off-policy Policy Gradient

- Policy Gradient: $\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{\xi \sim p_\theta}[R_{\geq t}(\xi) \nabla_\theta \log \pi_\theta(a_t | s_t)]$
- Off-Policy PG: $\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{\xi \sim p_{\theta'}}[\rho_{\theta'}^{\theta}(\xi_{\leq t}) R_{\geq t}(\xi) \nabla_\theta \log \pi_\theta(a_t | s_t)]$
  - $R_{\geq t}(\xi)$ = future discounted rewards affected by $\pi_\theta(a_t | s_t)$
  - $\rho_{\theta'}^{\theta}(\xi_{\leq t})$ = past probability ratios that affect $\pi_\theta(a_t | s_t)$



- Should we discount by γ^t? Not if we care about evidence from later states
- $\rho_{\theta'}^{\theta}(\xi_{\leq t})$ has high variance, some methods just use $\rho_{\theta'}^{\theta}(a_t | s_t) = \dfrac{\pi_\theta(a_t | s_t)}{\pi_{\theta'}(a_t | s_t)}$

MF

θ

DP

π'

max

[Liu et al., 2018]

Performance Difference Lemma

- Policy gradient = small changes in policy; can we make large changes?
- For any π, ξ: $\sum_t \gamma^t A_\pi(s_t, a_t) = \sum_t \gamma^t (r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)) = R(\xi) - V_\pi(s_0)$

  advantage of entire trajectory

- Expectation by different policy: Performance Difference Lemma

$$\sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\pi}[A_{\bar{\pi}}(s_t, a_t)] = \mathbb{E}_{\xi\sim p_\pi}[R(\xi) - V_{\bar{\pi}}(s_0)] = J_\pi - J_{\bar{\pi}}$$

$s_0 \sim p$ in both π and π'

  - We want to maximize over π, with $\bar{\pi}$ fixed

- Compare: PG Theorem $\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\theta}[A_{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t)]$

[Kakade and Langford, 2002]

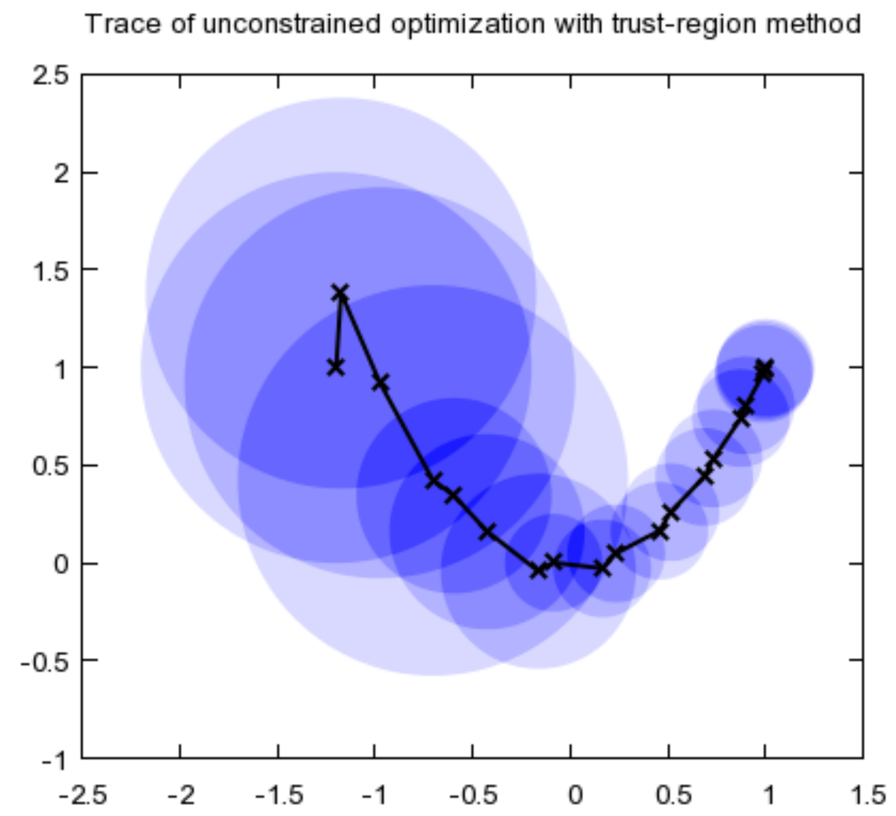
Finding best next policy

- With current policy $\bar{\pi}$: find $\max_\pi J_\pi - J_{\bar{\pi}} = \max_\pi \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\pi}[A_{\bar{\pi}}(s_t, a_t)]$
  - Can use $\bar{\pi}$ to evaluate $A_{\bar{\pi}}$
- But we don't have data $(s_t, a_t) \sim p_\pi$; idea: sample from $\bar{\pi}$
  - Trick question: is this on-policy or off-policy? On-policy data, but needs IS weight

$$\max_\pi \sum_t \gamma^t \mathbb{E}_{\xi_{\leq t}\sim p_{\bar{\pi}}}[\rho^{\pi}_{\bar{\pi}}(\xi_{\leq t}) A_{\bar{\pi}}(s_t, a_t)]$$

- Is it reasonable to use $\rho^{\pi}_{\bar{\pi}}(a_t | s_t) = \dfrac{\pi(a_t | s_t)}{\bar{\pi}(a_t | s_t)}$ instead? i.e. drop $\rho^{\pi}_{\bar{\pi}}(\xi_{<t})$

Trust-Region Policy Optimization (TRPO)



- Trust region = space around $\bar{\pi}$ where $\rho(\xi_{<t}) \approx 1$
  - Easier to consider $\mathbb{E}_{\xi_{<t} \sim p_{\bar{\pi}}}[\log \rho(\xi_{<t})] \approx 0$

$$-\mathbb{E}_{\xi_{<t} \sim p_{\bar{\pi}}}[\log \rho(\xi_{<t})] = \mathbb{D}[\bar{\pi}(\xi_{<t}) \| \pi(\xi_{<t})] = \sum_{t'<t} \mathbb{E}_{\xi_{<t'} \sim p_{\bar{\pi}}}[\mathbb{D}[\bar{\pi}(a_{t'} | s_{t'}) \| \pi(a_{t'} | s_{t'})]]$$

- TRPO: $\max_{\theta} \mathbb{E}_{(s,a) \sim p_{\bar{\theta}}}[\rho_{\bar{\theta}}^{\theta}(a | s) A_{\bar{\theta}}(s, a)]$ s.t. $\mathbb{E}_{s \sim p_{\bar{\theta}}}[\mathbb{D}[\pi_{\bar{\theta}}(a | s) \| \pi_{\theta}(a | s)]] \leq \epsilon$
  - $A_{\bar{\theta}}$ estimated with critic A_{ϕ}
  - Computational tricks for gradient-based optimization

MF

θ

DP

π'

max

[Schulman et al., 2015]

Proximal Policy Optimization (PPO)

MF

θ

DP

π'

max

- Same motivation: ascend $\mathbb{E}_{(s,a)\sim p_{\bar\theta}}[\rho_{\bar\theta}^{\theta}(a\,|\,s)A_{\bar\theta}(s,a)]$ with π_θ staying near $\pi_{\bar\theta}$
  - PPO-Penalty: add a penalty term for $\mathbb{E}_{s\sim p_{\bar\theta}}[\mathbb{D}[\pi_{\bar\theta}(a\,|\,s)\|\pi_\theta(a\,|\,s)]]$
  - PPO-Clip: ascend $\mathbb{E}_{(s,a)\sim p_{\bar\theta}}[L_{\bar\theta}^{\theta}(s,a)]$ with

$$L_{\bar\theta}^{\theta}(s,a) = \min(\rho_{\bar\theta}^{\theta}(a\,|\,s)A_{\bar\theta}(s,a), A_{\bar\theta}(s,a) + |\,\epsilon A_{\bar\theta}(s,a)\,|\,)$$

- This caps the incentive for θ to deviate from $\bar\theta$ at:
  - $\rho_{\bar\theta}^{\theta}(a\,|\,s) \leq 1+\epsilon$ for $A_{\bar\theta}(s,a) \geq 0$ (when we want to increase $\pi_\theta(a\,|\,s)$)
  - $\rho_{\bar\theta}^{\theta}(a\,|\,s) \geq 1-\epsilon$ for $A_{\bar\theta}(s,a) \leq 0$(when we want to decrease $\pi_\theta(a\,|\,s)$)

no incentive ≠ doesn't happen
PPO has lots more tricks
to limit divergence

[Schulman et al., 2017]

Recap

- Model-based policy evaluation can be solved linearly
- Deep RL isn't just SGD
  - Exception: policy gradient on offline (batch) data
- Value-based methods struggle to $\max$ in continuous action spaces
  - DDPG: π_θ learns to maximize Q_ϕ (actor–critic method)
- Importance Sampling decouples expectation and sampling distributions
  - Optimize on-policy objectives with off-policy data
  - TRPO and PPO: sample from current policy to evaluate next policy, if it's close