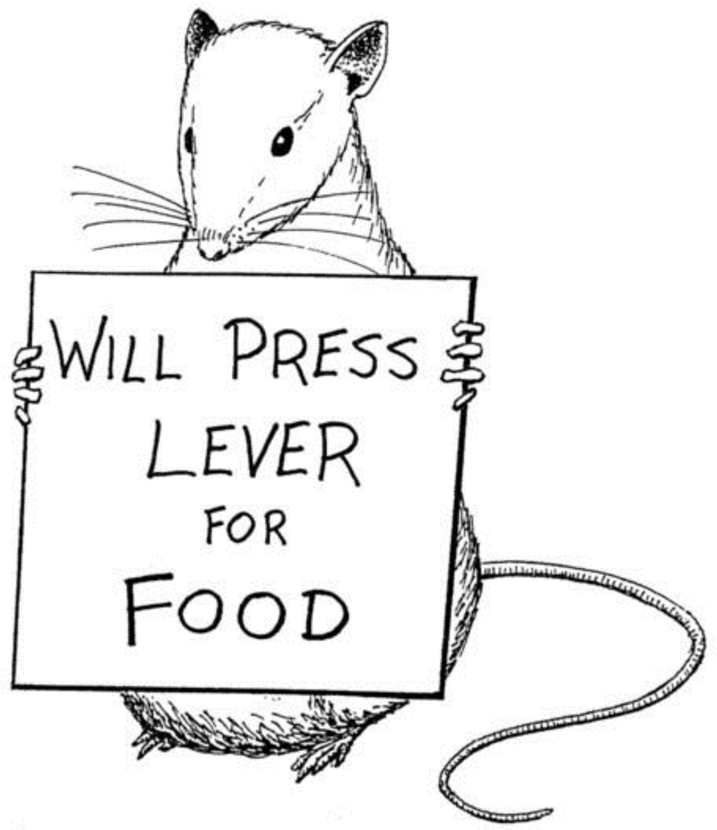
UCI University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2021

Lecture 6: Advanced Model-Free Methods

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



Recap

- Marginal state distributions can be computed recursively forward

$$p_\pi(s') = \mathbb{E}_{s \sim p_\pi}[\mathbb{E}_{a|s \sim \pi}[p(s' | s, a)]]$$

- Value functions can be computed recursively backward

$$V_\pi(s) = \mathbb{E}_{a|s \sim \pi}[r(s, a) + \gamma \mathbb{E}_{s'|s,a \sim p}[V_\pi(s')]]$$

- Forward and backward recursions are a recurring theme...

Recap: policy evaluation

	model-based	model-free
Monte Carlo (MC)	$V_\pi(s_0) = \mathbb{E}_{\xi\sim p_\pi}[R \mid s_0]$	$\xi \sim p_\pi \qquad V(s_0) \to R(\xi)$
Dynamic Programming (DP) / Temporal Difference (TD) (on-policy)	$V_\pi(s) = \mathbb{E}_{a\|s\sim\pi}[r(s,a) + \gamma\mathbb{E}_{s'\|s,a\sim p}[V_\pi(s')]]$	$s, a, r, s' \sim p_\pi \qquad V(s) \to r + \gamma V(s')$
Dynamic Programming (DP) / Temporal Difference (TD) (off-policy)	$Q_\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{\substack{s'\|s,a\sim p \\ a'\|s'\sim\pi}}[Q_\pi(s',a')]]$	$s, a, r, s' \sim p_{\pi'} \quad Q(s,a) \to r + \gamma\mathbb{E}_{a'\|s'\sim\pi}[Q(s',a')]$

Recap: policy ~~evaluation~~ improvement

	model-based	model-free
Monte Carlo (MC)	$V_\pi(s_0) = \mathbb{E}_{\xi\sim p_\pi}[R \mid s_0]$	$\xi \sim p_\pi \qquad V(s_0) \to R(\xi)$
Dynamic Programming (DP) / Temporal Difference (TD) (on-policy)	$V_\pi(s) = \overset{\max\limits_{a}}{\cancel{\mathbb{E}_{a\mid s\sim\pi}}}[r(s,a) + \gamma\mathbb{E}_{s'\mid s,a\sim p}[V_\pi(s')]]$ **Value Iteration**	$s, a, r, s' \sim p_\pi \qquad V(s) \to r + \gamma V(s')$
Dynamic Programming (DP) / Temporal Difference (TD) (off-policy)	$Q_\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{\substack{s'\mid s,a\sim p\\ \cancel{a'\mid s'\sim\pi}\\ \max\limits_{a'}}}[Q_\pi(s',a')]]$	$s, a, r, s' \sim p_{\pi'} \qquad Q(s,a) \to r + \gamma\overset{\max\limits_{a'}}{\cancel{\mathbb{E}_{a'\mid s'\sim\pi}}}[Q(s',a')]$ **Q-learning**

Recap: on- vs. off-policy

- On-policy:
  - We collect new data when policy changes
  - We quickly stop sampling old data
- Off-policy:
  - We use old data (or offline data) well after policy changes
- All optimizers must eventually train with support of their output policy
  - "On-policy optimizers" degrade with off-policy data
  - "Off-policy optimizers" improve with off-policy data, but saturate

Today's lecture

Bellman operator

Continuous action spaces

Off-policy evaluation, TRPO

Bellman operator

- Bellman operator:

$$\mathscr{B}[V](s) = \max_a \mathbb{E}[r + \gamma V(s') \,|\, s, a]$$

- Value Iteration = iteratively applying $\mathscr{B}$

- Why is this guaranteed to converge? $\mathscr{B}$ is a contraction:

$$\|\mathscr{B}[V_1] - \mathscr{B}[V_2]\|_\infty = \max_{s,a} \mathbb{E}[\gamma(V_1(s') - V_2(s')) \,|\, s, a] \leq \gamma \| V_1(s') - V_2(s')\|_\infty$$

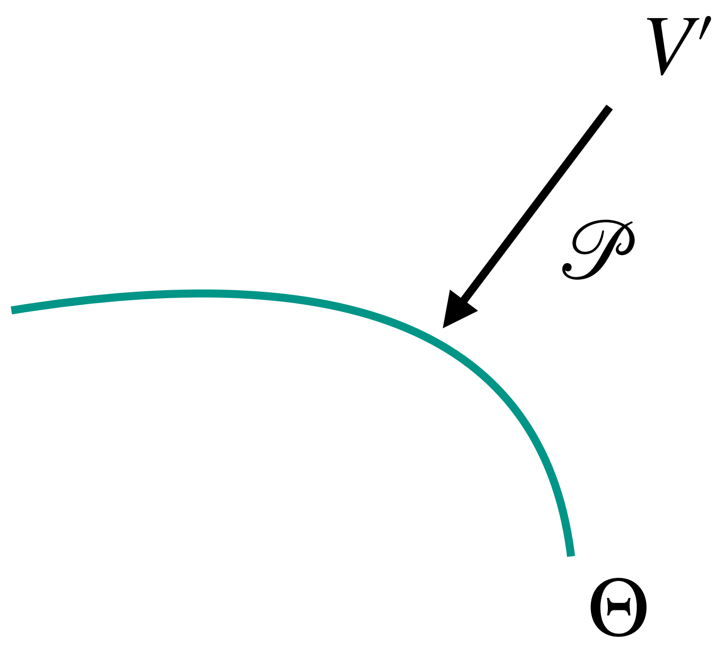
- $V^* = \mathscr{B}[V^*]$ is the unique fixed point

Fitted Value Iteration

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

- 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

$$\|\mathscr{P}[V'_1] - \mathscr{P}[V'_2]\|_2^2 \leq \|V'_1 - V'_2\|_2^2$$

- But the norms mismatch ($\mathscr{B}$: L_∞; $\mathscr{P}$: L_2)
  - So $\mathscr{P}\mathscr{B}$ is generally not a contraction

But isn't DQN just SGD?

Algorithm 1 DQN

```
initialize $\theta$ for $Q_\theta$, set $\bar\theta \leftarrow \theta$
for each step do
    if new episode, reset to $s_0$
    observe current state $s_t$
    take $\epsilon$-greedy action $a_t$ based on $Q_\theta(s_t, \cdot)$
        $\pi(a_t|s_t) = \begin{cases} 1 - \frac{|\mathcal{A}|-1}{|\mathcal{A}|}\epsilon & a_t = \operatorname{argmax}_a Q_\theta(s_t, a) \\ \frac{1}{|\mathcal{A}|}\epsilon & \text{otherwise} \end{cases}$
    get reward $r_t$ and observe next state $s_{t+1}$
    add $(s_t, a_t, r_t, s_{t+1})$ to replay buffer $\mathcal{D}$
    for each $(s, a, r, s')$ in minibatch sampled from $\mathcal{D}$ do
        $y \leftarrow \begin{cases} r & \text{if episode terminated at } s' \\ r + \gamma \max_{a'} Q_{\bar\theta}(s', a') & \text{otherwise} \end{cases}$
        compute gradient $\nabla_\theta (y - Q_\theta(s, a))^2$
    take minibatch gradient step
    every $K$ steps, set $\bar\theta \leftarrow \theta$
```

not exactly SGD

Is PG just SGD?

- Yes, inside the data collection loop

- But:

Algorithm 1 Actor–Critic

```
get on-policy sample $(s, a, r, s')$
take gradient step on $\mathcal{L}_\phi = (r + \gamma V_{\bar{\phi}}(s') - V_\phi(s))^2$
compute $\hat{A}(s, a) = r + \gamma V_\phi(s') - V_\phi(s)$
take gradient step $\nabla_\theta \log \pi_\theta(a|s) \hat{A}(s, a)$
repeat
```

Backup operator $\mathscr{B}_\pi[V] = \mathbb{E}_{a|s\sim\pi}[r + \gamma V(s') \,|\, s]$

- The critic's policy evaluation is not pure SGD

not to be confused with Bellman operator

$$\mathscr{B}[V](s) = \max_a \mathbb{E}[r + \gamma V(s') \,|\, s, a]$$

- No convergence guarantees (not even local!)

Exponential target updating

- Updating the target network every K iterations: $\bar{\theta}_i = \theta_{K\left\lfloor \frac{i}{K} \right\rfloor}$

- Using “fresher” target network (small K) reduces bias
  - But may destabilize the learning process

- Can we make the effective freshness the same for all gradient steps?

$$\bar{\theta}_i = \bar{\alpha} \sum_j (1 - \bar{\alpha})^j \theta_{i-j}$$

- Update $\bar{\theta} \leftarrow (1 - \bar{\alpha})\bar{\theta} + \bar{\alpha}\theta$ every step
  - With $\bar{\alpha} \approx \frac{1}{K}$

Today's lecture

Bellman operator

Continuous action spaces

Off-policy evaluation, TRPO

Continuous actions spaces

can do for large / continuous action spaces?

- What do we need for policy-based / actor–critic 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 $\mathscr{A}$, let's sample $a_1, \dots, a_k$ and take $\max_i Q(s, a_i)$
    - Sample from what distribution?
- Let's find an ad-hoc approximately greedy policy π
    - Sample $a_1, \dots, a_k$ from π
    - Take top $\frac{k}{c}$ “elite” samples
    - Fit π to the elites
    - Repeat

Idea 2: easily maximizable Q

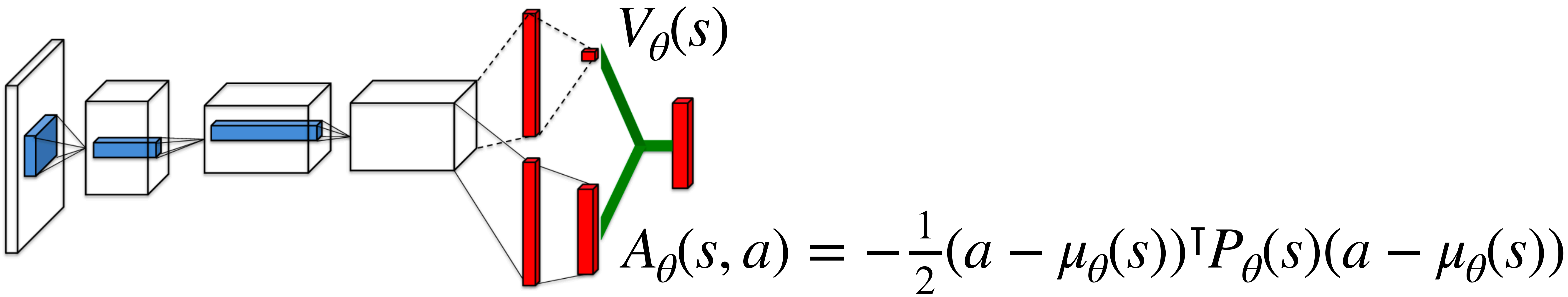
- Represent Q_θ in a way that is directly maximizable

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

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

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

- Architecture: dueling network


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

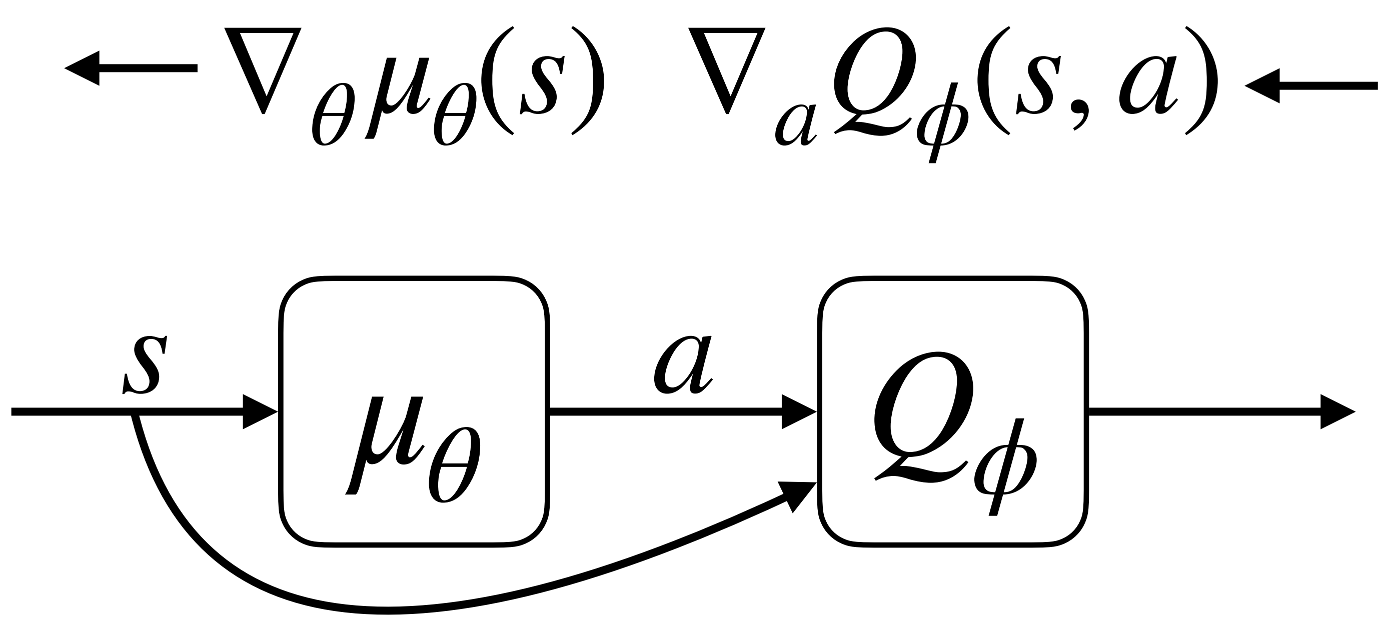
Idea 3: DDPG

- 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
- Policy Gradient Theorem:

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}_{s,a\sim p_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q_{\pi_\theta}(s,a)]$$

- Deterministic Policy Gradient Theorem:

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}_{s\sim p_\theta}\left[\nabla_\theta \mu_\theta(s)\, \nabla_a Q_{\mu_\theta}(s,a)\Big|_{a=\mu_\theta(s)}\right]$$



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

Today's lecture

Bellman operator

Continuous action spaces

Off-policy evaluation, TRPO

n-step DQN

- Instead of $y^1(r_t, s_{t+1}) = r_t + \gamma \max_{a_{t+1}} Q_{\bar{\theta}}(s_{t+1}, a_{t+1})$
- Take $y^n(r_t, \ldots, r_{t+n-1}, s_{t+n}) = r_t + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n \max_{a_{t+n}} Q_{\bar{\theta}}(s_{t+n}, a_{t+n})$
- Problem: $a_{t+1}, \ldots, a_{t+n-1}$ must all be on-policy
- Solutions:
  - Ignore the problem
  - Importance Sampling

Off-policy policy evaluation

- How to get an unbiased estimator of $\mathcal{J}_\theta = \mathbb{E}_{\xi \sim p_\theta}[R(\xi)]$

  from data sampled from a different distribution $\xi_1, \ldots, \xi_N \sim p_{\theta'}$?

$$\mathcal{J}_\theta = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\frac{p_\theta(\xi)}{p_{\theta'}(\xi)} R(\xi)\right]$$

$$\frac{p_\theta(\xi)}{p_{\theta'}(\xi)} = \prod_t \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta'}(a_t \mid s_t)}$$

- A reward r_t is not affected by future divergence

$$\mathcal{J}_\theta = \sum_t \mathbb{E}_{s_t, a_t \sim p_{\theta'}}\left[\gamma^t r_t \prod_{t' \leq t} \frac{\pi_\theta(a_{t'} \mid s_{t'})}{\pi_{\theta'}(a_{t'} \mid s_{t'})}\right]$$

Off-policy Policy Gradient

$$\mathcal{J}_\theta = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\frac{p_\theta(\xi)}{p_{\theta'}(\xi)} R(\xi)\right]$$

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\frac{\nabla_\theta p_\theta(\xi)}{p_{\theta'}(\xi)} R(\xi)\right]$$

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\frac{\nabla_\theta p_\theta(\xi)}{p_{\theta'}(\xi)} R(\xi)\right] = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\frac{p_\theta(\xi)}{p_{\theta'}(\xi)} \nabla_\theta \log p_\theta(\xi) R(\xi)\right]$$

$$= \mathbb{E}_{\xi \sim p_{\theta'}}\left[\prod_t \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)} \sum_{t'} \nabla_\theta \log \pi_\theta(a_{t'}|s_{t'}) \sum_{t''} \gamma^{t''} r_{t''}\right]$$

forward

$$= \mathbb{E}_{\xi \sim p_{\theta'}}\left[\sum_{t'} \prod_{t \leq t'} \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)} \nabla_\theta \log \pi_\theta(a_{t'}|s_{t'}) \sum_{t'' \geq t'} \gamma^{t''} r_{t''}\right]$$

backward

Off-policy Policy Gradient: approximation

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}_{\xi\sim p_{\theta'}}\left[\sum_{t'}\prod_{t\leqslant t'}\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}\nabla_\theta \log \pi_\theta(a_{t'}|s_{t'})\sum_{t''\geqslant t'}\gamma^{t''}r_{t''}\right]$$

$$= \sum_{t'}\mathbb{E}_{s_{t'},a_{t'}\sim p_{\theta'}}\left[C_{\theta,\theta',t'}\frac{\pi_\theta(a_{t'}|s_{t'})}{\pi_{\theta'}(a_t|s_t)}\nabla_\theta \log \pi_\theta(a_{t'}|s_{t'})\hat{A}_t\right]$$

- $C_{\theta,\theta',t'}$ is the important sampling coefficient of past actions, marginalized
  - Originally just ignored ¯_(ツ)_/¯

More analysis

$$\sum_t \gamma^t \hat{A}^1_{\pi_\theta}(s_t, a_t) = \sum_t \gamma^t (r(s_t, a_t) + \gamma V_{\pi_\theta}(s_{t+1}) - V_{\pi_\theta}(s_t))$$

$$= \sum_t \gamma^t r(s_t, a_t) - V_{\pi_\theta}(s_0)$$

$$\mathbb{E}_{\xi \sim p_{\theta'}}\left[\sum_t \gamma^t \hat{A}^1_{\pi_\theta}(s_t, a_t)\right] = \mathbb{E}_{\xi \sim p_{\theta'}}\left[\sum_t \gamma^t r(s_t, a_t) - V_{\pi_\theta}(s_0)\right] = \mathcal{J}_{\theta'} - \mathcal{J}_\theta$$

$$= \sum_t \gamma^t \, \mathbb{E}_{s_t, a_t \sim p_{\theta'}}[\hat{A}^1_{\pi_\theta}(s_t, a_t)]$$

$$= \sum_t \gamma^t \, \mathbb{E}_{s_t \sim p_{\theta'}}\left[\mathbb{E}_{a_t | s_t \sim \pi_\theta}\left[\frac{\pi_{\theta'}(a_t | s_t)}{\pi_\theta(a_t | s_t)} \hat{A}^1_{\pi_\theta}(s_t, a_t)\right]\right]$$

- Can we switch to $s_t \sim p_\theta$, so we can estimate the expectation empirically?

Trust-Region Policy Optimization (TRPO)

$$\max_{\theta'} \sum_t \gamma^t \, \mathbb{E}_{s_t \sim p_\theta} \left[\mathbb{E}_{a_t|s_t \sim \pi_\theta} \left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \hat{A}^1_{\pi_\theta}(s_t, a_t) \right] \right]$$

$$\text{s.t.} \quad \mathbb{D}[\pi_{\theta'} \| \pi_\theta] \leqslant \epsilon$$

- For small ϵ, the objective is close to $\mathscr{J}_{\theta'} - \mathscr{J}_\theta$
  - Guarantees improvement

$$\mathcal{L}_\theta(s, a, r, s') = -\frac{\pi_\theta(a|s)}{\pi_{\bar{\theta}}(a|s)}(r + \gamma V_\phi(s') - V_\phi(s)) + \lambda(\mathbb{D}[\pi_\theta(\cdot|s) \| \pi_{\bar{\theta}}(\cdot|s)] - \epsilon)$$

Recap

- Deep RL isn't just SGD
  - Except for the purest PG — which has high variance of the gradient estimator
- In continuous action spaces, policy should probably be represented
- Importance-sampling methods for off-policy
  - Challenging to do exactly, so we use heuristic approximations