UCI University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2021

Lecture 5: Actor–Critic Methods

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

Logistics

assignments

- Assignment 1 due this Friday
- Assignment 2 to be published soon
  - Due next Friday

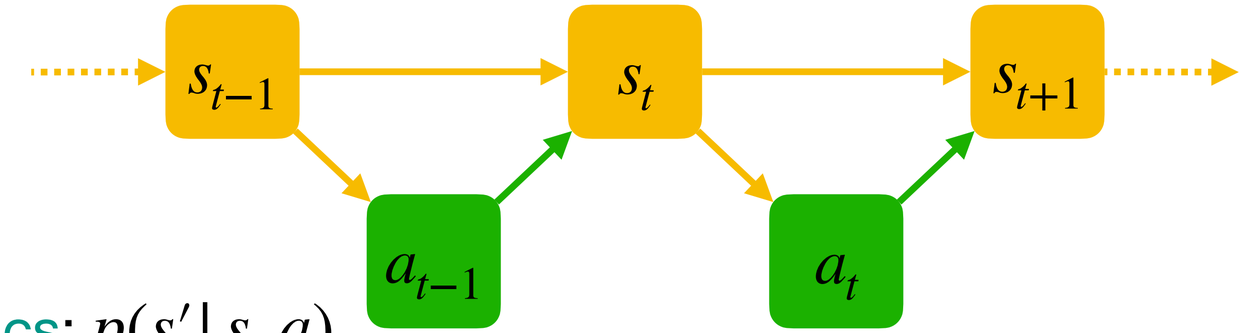
Recap

- Stochastic process = sequence of random variables $x_0, x_1, x_2, \ldots$
- The probability of a realization of the variables is given by
- $$p(x_0, x_1, x_2, \ldots) = p(x_0)p(x_1 \,|\, x_0)p(x_2 \,|\, x_0, x_1)\cdots = \prod_i p(x_{i+1} \,|\, x_{\leq i})$$
- Often we have structure that simplifies these probabilities
- E.g., the Markov property $p(x_{i+1} \,|\, x_{\leq i}) = p(x_{i+1} \,|\, x_i)$
  - x_{i+1} is independent of $x_0, \ldots, x_{i-1}$ given x_i

Recap

- Dynamical system = stochastic process over state s_t
  - For now we assume Markov dynamics $p(s_{t+1} \mid s_t)$
  - The initial state distribution $p(s_0)$ also considered part of the dynamics
- The dynamics induces a joint distribution $p(s_0, s_1) = p(s_0)p(s_1 \mid s_0)$
- The dynamics also induces a marginal distribution $p(s_1) = \sum_{s_0} p(s_0, s_1)$
  - More generally, we have the recursion: $p(s') = \sum_{s} p(s)p(s' \mid s)$

Recap



- Controlled dynamics: $p(s' \mid s, a)$
- A given control policy $\pi(a \mid s)$ induces the dynamics
- $p_\pi(s' \mid s) = \sum_a \pi(a \mid s) p(s' \mid s, a)$
- Can similarly compute marginals recursively $p_\pi(s') = \sum_{s,a} p_\pi(s) \pi(a \mid s) p(s' \mid s, a)$
  - and $p_\pi(s', a') = \sum_{s,a} p_\pi(s, a) p(s' \mid s, a) \pi(a' \mid s')$

Recap

- Trajectory: $\xi = s_0, a_0, s_1, a_1, s_2, \ldots$
- Probability of a trajectory: $p_\pi(\xi) = p(s_0) \prod_t \pi(a_t \,|\, s_t) p(s_{t+1} \,|\, s_t, a_t)$
- Marginal expectation:

$$\mathbb{E}_{\xi \sim p_\pi}[f(s_t)] = \sum_\xi p_\pi(\xi) f(s_t)$$

Recap

- Trajectory: $\xi = s_0, a_0, s_1, a_1, s_2, \ldots$
- Probability of a trajectory: $p_\pi(\xi) = p(s_0) \prod_t \pi(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t)$
- Marginal expectation:

$$\begin{aligned}
\mathbb{E}_{\xi \sim p_\pi}[f(s_t)] &= \sum_\xi p_\pi(\xi) f(s_t) \\
&= \mathbb{E}_{s_0 \sim p}[\mathbb{E}_{a_0 \mid s_0 \sim \pi}[\cdots \mathbb{E}_{s_t \mid s_{t-1}, a_{t-1} \sim p}[\cdots \mathbb{E}_{s_T \mid s_{T-1}, a_{T-1} \sim p}[f(s_t)] \cdots] \cdots]] \\
&= \mathbb{E}_{s_0 \sim p}[\mathbb{E}_{a_0 \mid s_0 \sim \pi}[\cdots \mathbb{E}_{s_t \mid s_{t-1}, a_{t-1} \sim p}[f(s_t)] \cdots]] \\
&= \mathbb{E}_{s_t \sim p_\pi}[\mathbb{E}_{\xi \mid s_t \sim p_\pi}[f(s_t)]] = \mathbb{E}_{s_t \sim p_\pi}[f(s_t)]
\end{aligned}$$

Rao–Blackwell theory

- Suppose we use data x to generate an estimate $\hat{\theta}(x)$ of parameter θ
- Let y be a sufficient statistic of x for θ
  - That is, there's nothing more, on top of y, that x can tell us about θ
- Consider the estimator $\hat{\theta}(y) = \mathbb{E}[\hat{\theta}(x) \,|\, y]$ of θ
  - It has the same bias as $\hat{\theta}(x)$, and lower variance
  - Which also means it has lower MSE

Don't let the past distract you

$$\nabla_\theta \mathscr{J}_\theta = \mathbb{E}_{\xi \sim p_\theta}\left[\left(\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right) R\right] = \sum_t \mathbb{E}_{s_t \sim p_\theta}[\nabla_\theta \mathbb{E}_{a_t \mid s_t \sim \pi_\theta}[R]]$$

- In our case, $R_{\geq t} = \sum_{t' \geq t} \gamma^{t'} r(s_{t'}, a_{t'})$ is a sufficient statistic of R for $\mathscr{J}_\theta$

- Therefore, a lower-variance gradient estimator:

$$\sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta}[\mathbb{E}_{a_t \mid s_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) R_{\geq t}]]$$

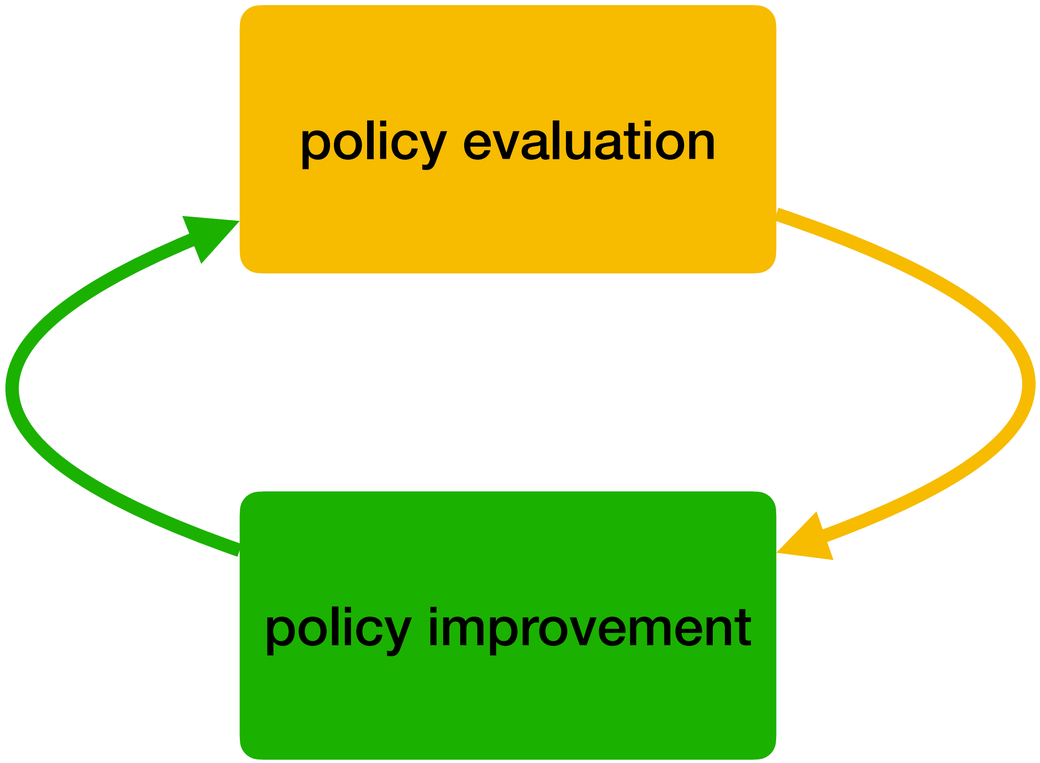
Today's lecture

PG with learned value function

Actor–Critic methods

Advantage estimation

Policy-based methods



e.g. MC

differentiable policy

Policy Gradient (PG) with reward-to-go

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

Policy Gradient (PG) with reward-to-go

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

$$= \mathbb{E}_{\xi \sim p_\theta}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) R\right]$$

$$= \sum_t \mathbb{E}_{s_t \sim p_\theta}[\mathbb{E}_{a_t \mid s_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) R]]$$

$$= \sum_t \mathbb{E}_{s_t \sim p_\theta}[\nabla_\theta \mathbb{E}_{a_t \mid s_t \sim \pi_\theta}[R]]$$

Assuming undiscounted horizon

$$= \sum_t \mathbb{E}_{s_t \sim p_\theta}\left[\nabla_\theta \mathbb{E}_{a_t \mid s_t \sim \pi_\theta}\left[\sum_{t'} r_{t'}\right]\right]$$

Gradient estimator is independent of past rewards

$$= \sum_t \mathbb{E}_{s_t \sim p_\theta}\left[\nabla_\theta \mathbb{E}_{a_t \mid s_t \sim \pi_\theta}\left[\sum_{t' \geq t} r_{t'}\right]\right]$$

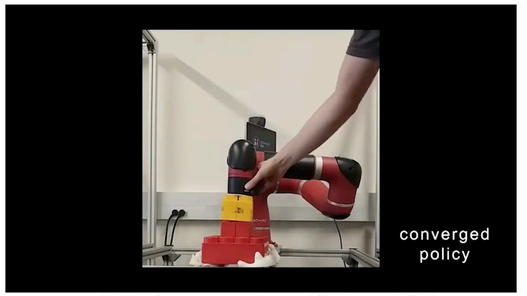
Discounted case

$$\nabla_\theta \mathcal{J}_\theta = \sum_t \mathbb{E}_{s_t \sim p_\theta} \left[\mathbb{E}_{a_t | s_t \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t' \geq t} \gamma^{t'} r_{t'} \right] \right]$$

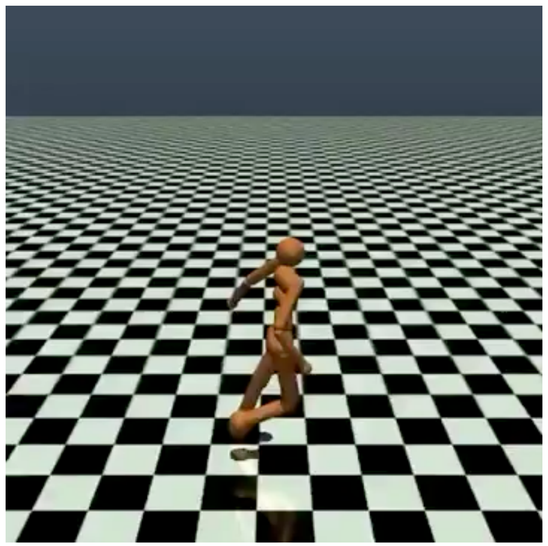
Discounted case

$$\nabla_\theta \mathcal{J}_\theta = \sum_t \mathbb{E}_{s_t \sim p_\theta} \left[\mathbb{E}_{a_t | s_t \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t' \geq t} \gamma^{t'} r_{t'} \right] \right]$$

$$= \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta} \left[\mathbb{E}_{a_t | s_t \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t' \geq t} \gamma^{t'-t} r_{t'} \right] \right]$$



- Does it make sense to discount the contribution of future states?
  - Do we really care less what we do long after the start?
- In a sense, discounting is not real, just a computational / statistical trick
  - Don't discount the contribution of s_t to the loss
  - That's what most algorithms do



Policy-Gradient Theorem

$$\nabla_\theta \mathcal{J}_\theta = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta}[\mathbb{E}_{a_t | s_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) R_{\geq t}]]$$

Policy-Gradient Theorem

$$\nabla_\theta \mathcal{J}_\theta = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta}[\mathbb{E}_{a_t | s_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) R_{\geq t}]]$$

$$\overset{!}{=} \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta}[\mathbb{E}_{a_t | s_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) Q_{\pi_\theta}(s_t, a_t)]]$$

$$\begin{aligned}
\nabla_\theta V_{\pi_\theta}(s) &= \nabla_\theta \mathbb{E}_{a | s \sim \pi_\theta}[Q_{\pi_\theta}(s, a)] \\
&= \sum_a (\nabla_\theta \pi_\theta(a \mid s) Q_{\pi_\theta}(s, a) + \pi_\theta(a \mid s) \nabla_\theta Q_{\pi_\theta}(s, a)) \\
&= \mathbb{E}_{a | s \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a \mid s) Q_{\pi_\theta}(s, a) + \nabla_\theta (r(s, a) + \gamma \mathbb{E}_{s' | s, a \sim p}[V_{\pi_\theta}(s')])] \\
&= \mathbb{E}_{a | s \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a \mid s) Q_{\pi_\theta}(s, a) + \gamma \mathbb{E}_{s' | s, a}[\nabla_\theta V_{\pi_\theta}(s')]]
\end{aligned}$$

- Here back-propagating gradients is like a Bellman recursion

Today's lecture

PG with learned value function

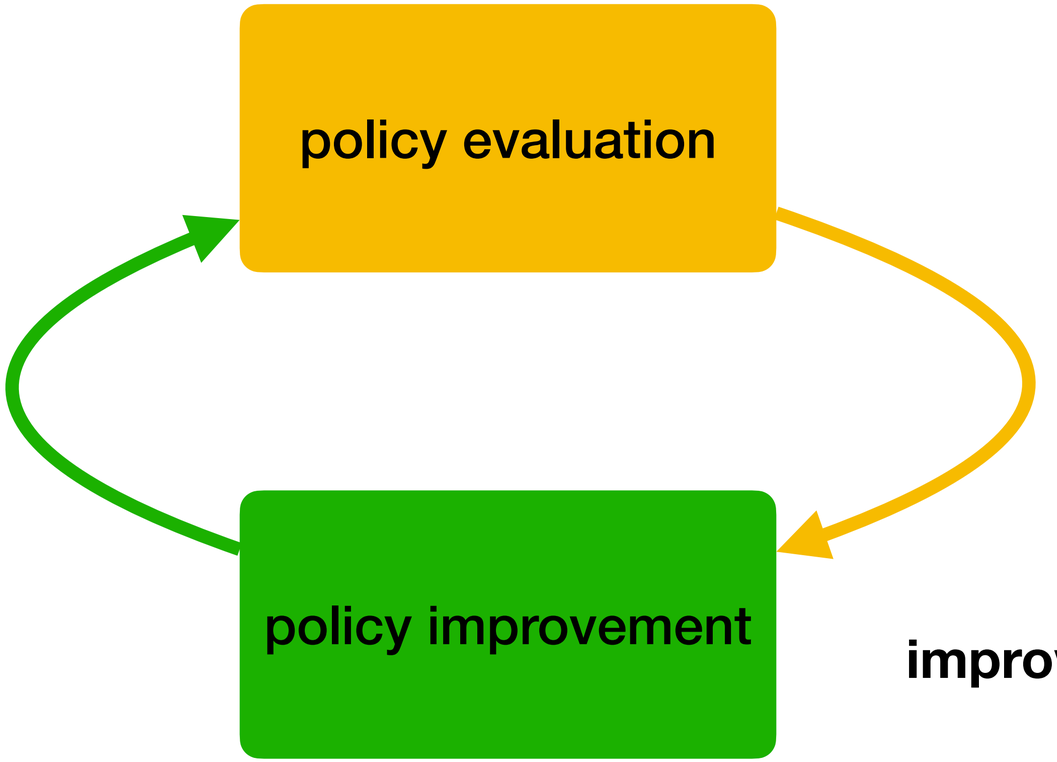
Actor–Critic methods

Advantage estimation

Actor–Critic methods

$$\mathscr{L}_\theta(s,a) = -\nabla_\theta \log \pi_\theta(a \mid s) Q_\phi(s,a)$$

$$\mathscr{L}_\phi(s,a,r,s') = (r + \gamma \mathbb{E}_{a' \mid s' \sim \pi_\theta}[Q_{\bar{\phi}}(s',a')] - Q_\phi(s,a))^2$$



critic:
evaluates the actor

actor:
improves using the critic's advice

Baselines

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}[\nabla_\theta \log \pi_\theta(a \mid s)(Q_{\pi_\theta}(s,a) - b) \mid s, a]$$

- b can be any variable independent of a given s
  - Can depend on the past, but not the future
- Previously, we used $b = \frac{1}{N} \sum_i R_i$
- This suggests using $b = V_{\pi_\theta}(s)$

Advantage estimation

$$\nabla_\theta \mathcal{J}_\theta = \mathbb{E}[\nabla_\theta \log \pi_\theta(a \mid s) A_{\pi_\theta}(s, a) \mid s, a]$$

- How to estimate $A_{\pi_\theta}(s, a)$?

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

- With value estimation $V_\phi(s)$, estimate the advantage:

$$\hat{A}(s, a) \approx r + \gamma V_\phi(s') - V_\phi(s)$$

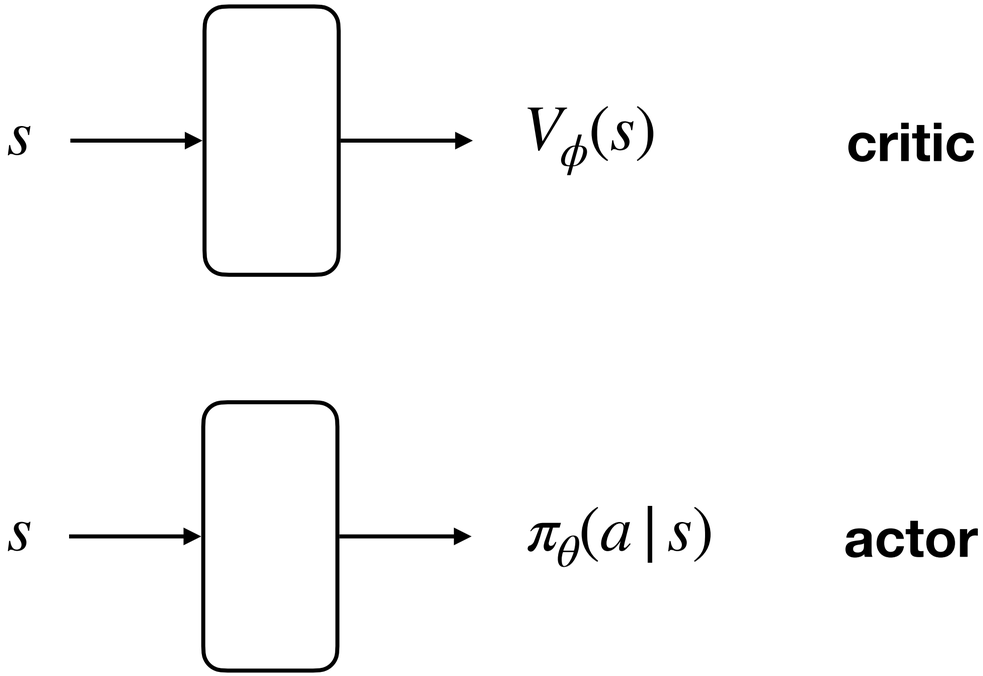
An Actor–Critic algorithm

Algorithm 1 Actor–Critic

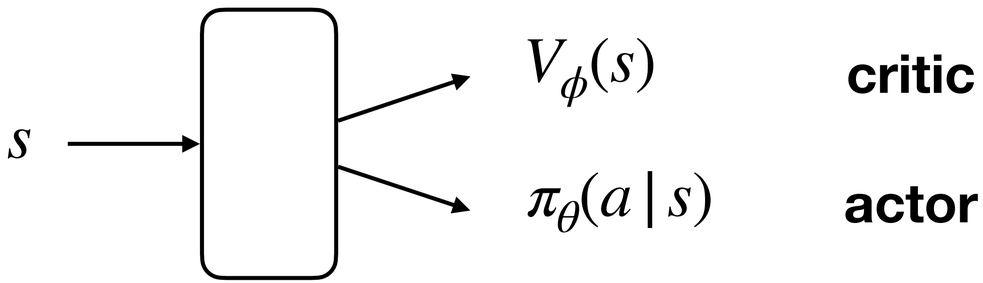
get on-policy sample (s, a, r, s')
take gradient step on $\mathcal{L}_\phi = (r + \gamma V_{\bar{\phi}}(s') - V_\phi(s))^2$
compute $\hat{A}(s, a) = r + \gamma V_\phi(s') - V_\phi(s)$
take gradient step $\nabla_\theta \log \pi_\theta(a|s)\hat{A}(s, a)$
repeat

Practical considerations: param sharing

- Separate parameters:

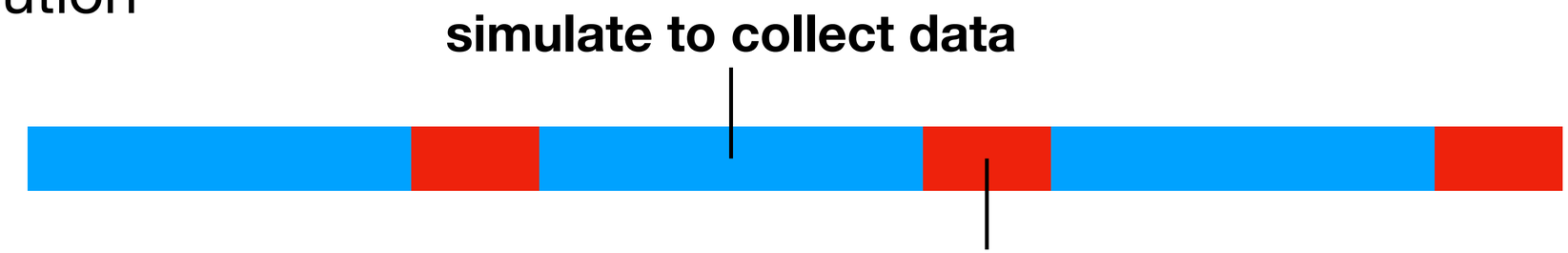


- Shared parameters:
  - Can be more data efficient
  - Can be less stable

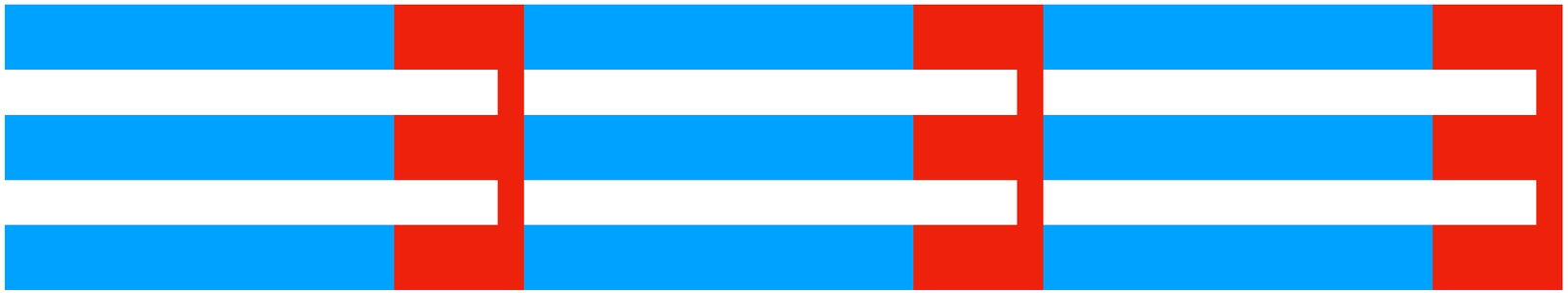


Practical considerations: distributed comp.

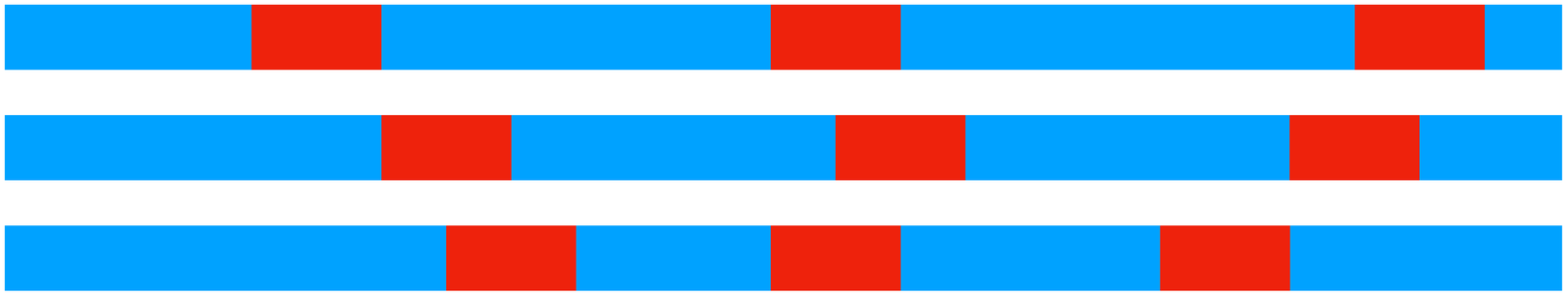
- Serial execution



- Synchronous parallel execution



- Asynchronous parallel execution



Today's lecture

PG with learned value function

Actor–Critic methods

Advantage estimation

More advantage estimation

$$\hat{A}(s_t, a_t) \approx \sum_{t' \geq t} \gamma^{t'-t} r_{t'} - V_\phi(s_t)$$

- Asynchronous Advantage Actor Critic (A3C):
  - MC advantage estimation + asynchronous parallel execution
- Advantage Actor Critic (A2C): same but serial

Comparing advantage estimators

	bias	variance
$\nabla_\theta \mathcal{J}_\theta \approx \nabla_\theta \log \pi_\theta(a \mid s) \left(\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - b \right)$	none	high one grad per traj
$\nabla_\theta \mathcal{J}_\theta \approx \nabla_\theta \log \pi_\theta(a \mid s)(r + \gamma V_\phi(s') - V_\phi(s))$	some approx value	lower
$\nabla_\theta \mathcal{J}_\theta \approx \nabla_\theta \log \pi_\theta(a \mid s) \left(\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - V_\phi(s) \right)$	none	mid state-dependent baseline

n-step TD

- 1-step TD: $\hat{A}_t^1 = r_t + \gamma V(s_{t+1}) - V(s_t)$
- 2-step TD: $\hat{A}_t^2 = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$

  ...

- n-step TD: $\hat{A}_t^n = r_t + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n V(s_{t+n}) - V(s_t)$
- In the limit (MC): $\hat{A}_t^\infty = r_t + \gamma r_{t+1} + \cdots - V(s_t)$

TD(λ)

- How to choose n?
- Any specific n is hard truncation of the window of evidence we consider
- Instead, use “exponential window”
  - Take n-step TD with weight proportional to λ^{n-1}, where $0 \leq \lambda \leq 1$
- Now with

$$\hat{A}_t^n = \sum_{t'=0}^{n-1} \gamma^{t'} \hat{A}_{t+t'}^1 = \sum_{t'=0}^{n-1} \gamma^{t'} (r_{t+t'} + \gamma V(s_{t+t'+1}) - V(s_{t+t'}))$$

TD(λ)

- How to choose n?
- Any specific n is hard truncation of the window of evidence we consider
- Instead, use “exponential window”
  - Take n-step TD with weight proportional to λ^{n-1}, where $0 \leq \lambda \leq 1$
- Now with

$$\hat{A}_t^n = \sum_{t'=0}^{n-1} \gamma^{t'} \hat{A}_{t+t'}^1 = \sum_{t'=0}^{n-1} \gamma^{t'} (r_{t+t'} + \gamma V(s_{t+t'+1}) - V(s_{t+t'}))$$

$$\hat{A}_t^\lambda = (1-\lambda) \sum_n \lambda^{n-1} \hat{A}_t^n = (1-\lambda) \sum_n \lambda^{n-1} \sum_{t'=0}^{n-1} \gamma^{t'} \hat{A}_{t+t'}^1$$

$$= (1-\lambda) \sum_{t'} \gamma^{t'} \hat{A}_{t+t'}^1 \sum_{n \geq t'+1} \lambda^{n-1} = \sum_{t'} (\lambda\gamma)^{t'} \hat{A}_{t+t'}^1$$

Generalized Advantage Estimation (GAE(λ))

$$\nabla_\theta \mathcal{J}_\theta \approx \nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \sum_{t'} (\lambda\gamma)^{t'} \hat{A}^1_{t+t'}$$

Generalized Advantage Estimation (GAE(λ))

$$\nabla_\theta \mathcal{J}_\theta \approx \nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \sum_{t'} (\lambda\gamma)^{t'} \hat{A}^1_{t+t'}$$

$$\hat{A}^1_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

- GAE(0) = 1-step; GAE(1) = MC

Recap

- Policy-Gradient Theorem
- State-dependent baselines
- Actor–Critic methods
  - Advantage estimation
  - Practical considerations
- n-step TD and TD(λ)