Off-policy learning II

Konpat Preechakul
Chulalongkorn University
September 2019

Previously …

Recap off-policy

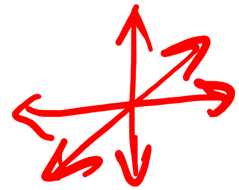
Behavior policy = any other policy $b(a|s)$

Target policy = our policy $\pi(a|s)$

On-policy = experience from target policy

Off-policy = experience from behavior policy

Recap overview



Model-free	Model-based
Environment is a black box	We know environment (transitions, rewards)
Value-based $q \to \pi$	**Policy-based** $\pi \leftarrow q$
We learn value. Use value to improve policy greedily	We directly learn policy (from some value)
On-policy	**Off-policy**
Experience comes from target policy (interactive experience)	Experience comes from behavior policy (observative experience)

Model-free vs Model-based

Model-free	Model-based
Monte Carlo TD (SARSA, Q-learning) N-step TD (n-step SARSA) TD(lambda)	Dynamic programming

Policy iteration, value iteration could be used on both sides

Value-based vs Policy-based

Value-based	Policy-based
Everything we have learned so far … Dynamic programming MC TD N-step TD	Policy gradient (future lectures)

On-policy vs Off-policy

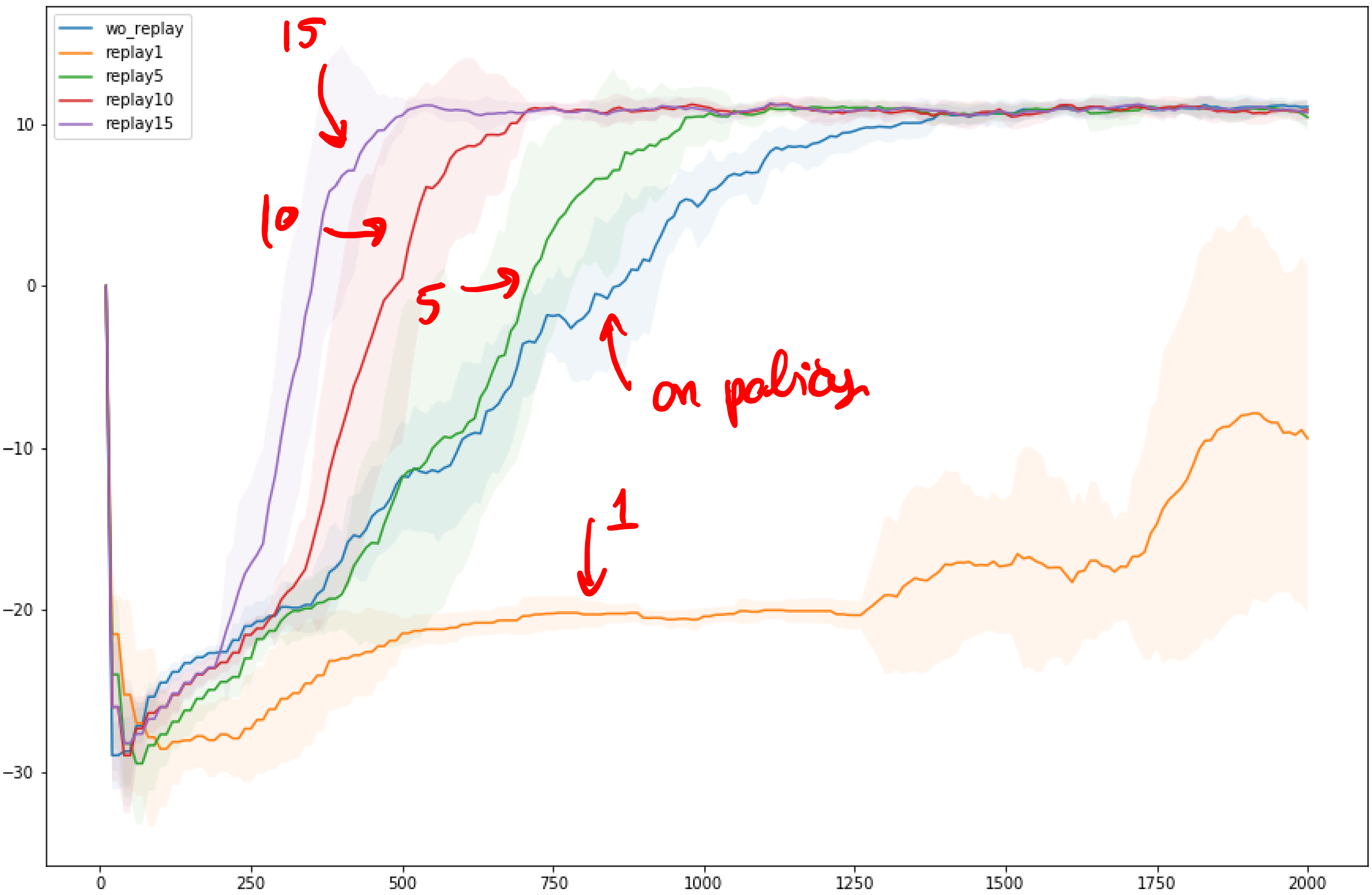
On-policy	Off-policy
MC SARSA N-step SARSA	Q-learning Expected SARSA (why?) Deterministic policies (why?)

Dynamic programming doesn't use experience to learn.
Not on this scale.

Off-policy also means on-policy.

take a $(s, a, r, s') \rightarrow \text{DB} \xrightarrow{n} (s, a, r, s') \times n$

Why do we want off-policy?



Recap importance sampling (IS)

Importance sampling ratio

$$\rho_{t:T-1} = \frac{\mathbb{P}^{\pi}(\tau)}{\mathbb{P}^{b}(\tau)}$$

$$\rho_{t:T-1} = \prod_{i=t}^{T-1} \frac{\pi(a_i|s_i)}{b(a_i|s_i)}$$

Value function becomes

behavior

$$v(s_t) = \mathbb{E}_b\left[\rho_{t:T-1} G_t | S_t = s_t\right] = \mathbb{E}_{\pi}$$

Importance sampling (IS)

$$\rho_{t:T-1} = \frac{\boxed{\mathbb{P}^\pi(\tau)}}{\mathbb{P}^b(\tau)} \Rightarrow \frac{2}{1}$$

Intuition

- If action is more likely on "target", IS > 1
- If action is less likely on "target", IS < 1
- What if target never takes action behavior takes?

Requirement

- Behavior must be "exploratory"
- Behavior policy is known $b(a|s)$

Why stochastic behavior policy?

$$\rho_{t:T-1} = \prod_{i=t}^{T-1} \frac{\pi(a_i|s_i)}{b(a_i|s_i)} \to \infty$$

$\to 0$

$b(a|s) > 0 \ \forall a$

$\Downarrow$

π

- If b(a|s) = 0 for some state s
- IS ratio => infinity
- We want to keep the ratio=1 on average
- The ratio will be 1 if behavior = target
- **Ratio = how informative is the behavior**

IS for deterministic policy

$\pi(a|s) = 1 \; ; a = 1$

$$\pi(a|s) = 0 \quad \text{almost everywhere}$$

$$p = \prod \frac{\cancel{\pi(a|s)}^{\,0}}{b(a|s)}$$

$$\frac{\pi(a|s)}{b(a|s)} = 0 \quad \text{almost everywhere}$$

- Usually behavior will give much “less” information for this kind of policy
- Learning becomes very slow

More on off-policy learning

Variance of importance sampling

- We need a stochastic behavior policy
- The variance could go to "infinity"

$$\mathrm{Var}(X) = \mathbb{E}\left[(X - \bar{X})^2\right] = \mathbb{E}\left[X^2 - 2X\bar{X} - \bar{X}^2\right]$$

$\rho G =$

$$= \mathbb{E}\left[X^2\right] - \bar{X}^2$$

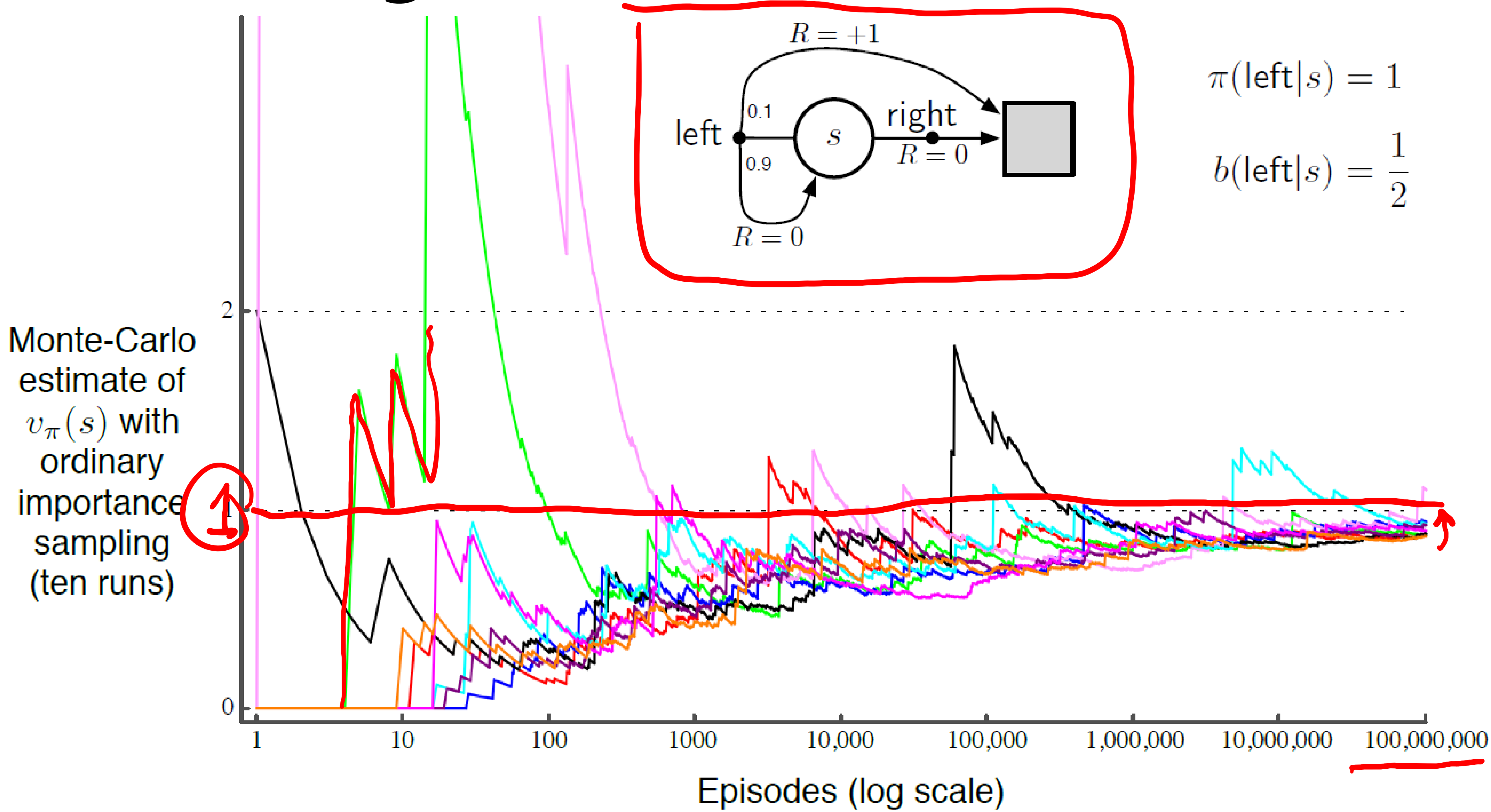
- If the $\mathbb{E}\left[X^2\right]$ goes to infinity, $\mathrm{Var}(X)$ goes to inf.

∞

- $\rho = \prod_{i=t}^{T-1} \frac{\pi(a_i|s_i)}{b(a_i|s_i)}$ could be infinity if T goes infinity

> 0

High variance hurts convergence (10 runs)



Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT press.

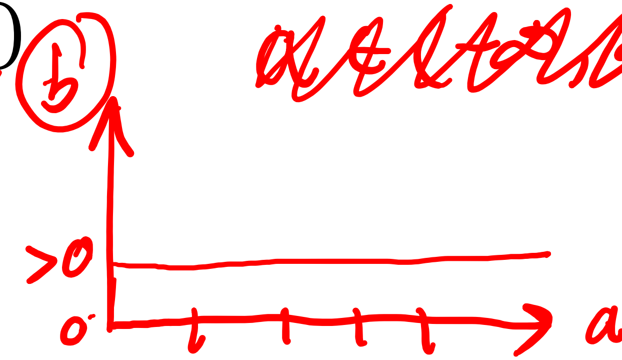
Okay, IS is bad, do we have alternatives?

A few steps IS might not be that bad?

N-step TD with IS

- This could reduce the effect of high variance
- If behavior policy is stochastic, the variance is bounded (discrete actions)

$$\prod_{i=0}^{n-1} \frac{\pi(a_{t+i}|s_{t+i})}{b(a_{t+i}|s_{t+i})} < \infty$$



- This could be extended to lambda (average of many n-step returns)

Recap N-step SARSA with IS

$(s_t, a_t)\ s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2})\quad a_{t+3}$

$$v(s_t) \leftarrow v(s_t) + \alpha \rho_{t:t+n-1} \left[G_{t:t+n} - v(s_t) \right]$$

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha \rho_{t+1:t+n} \left[G_{t:t+n} - q(s_t, a_t) \right]$$

- The first action is "given" to action-value function
  - No need for correction
- Additional "last" action is sampled in action-value function
  - Need for correction

How about no IS at all?

Surprisingly there are a few algorithms which are just fine without importance sampling

Expected SARSA

$q(s_{t+1}, a_{t+1})$

$$q(s_t, a_t) \leftarrow$$
$$q(s_t, a_t) + \alpha\left[r(s_t, a_t) + \mathbb{E}_{a\sim\pi}[q(s_{t+1}, a)] - q(s_t, a_t)\right]$$

- We use "expectation" to correct for the behavior policy
- Requires only (s, a, r, s') a' ×
- (s, a) are given
- (r, s') are independent from policy given (s, a)
- The same reason for Q-learning

Deterministic policies

Q-learning

$\pi(s) = a$

- The expectation has a simple form

π a'

(s, a, r, s')

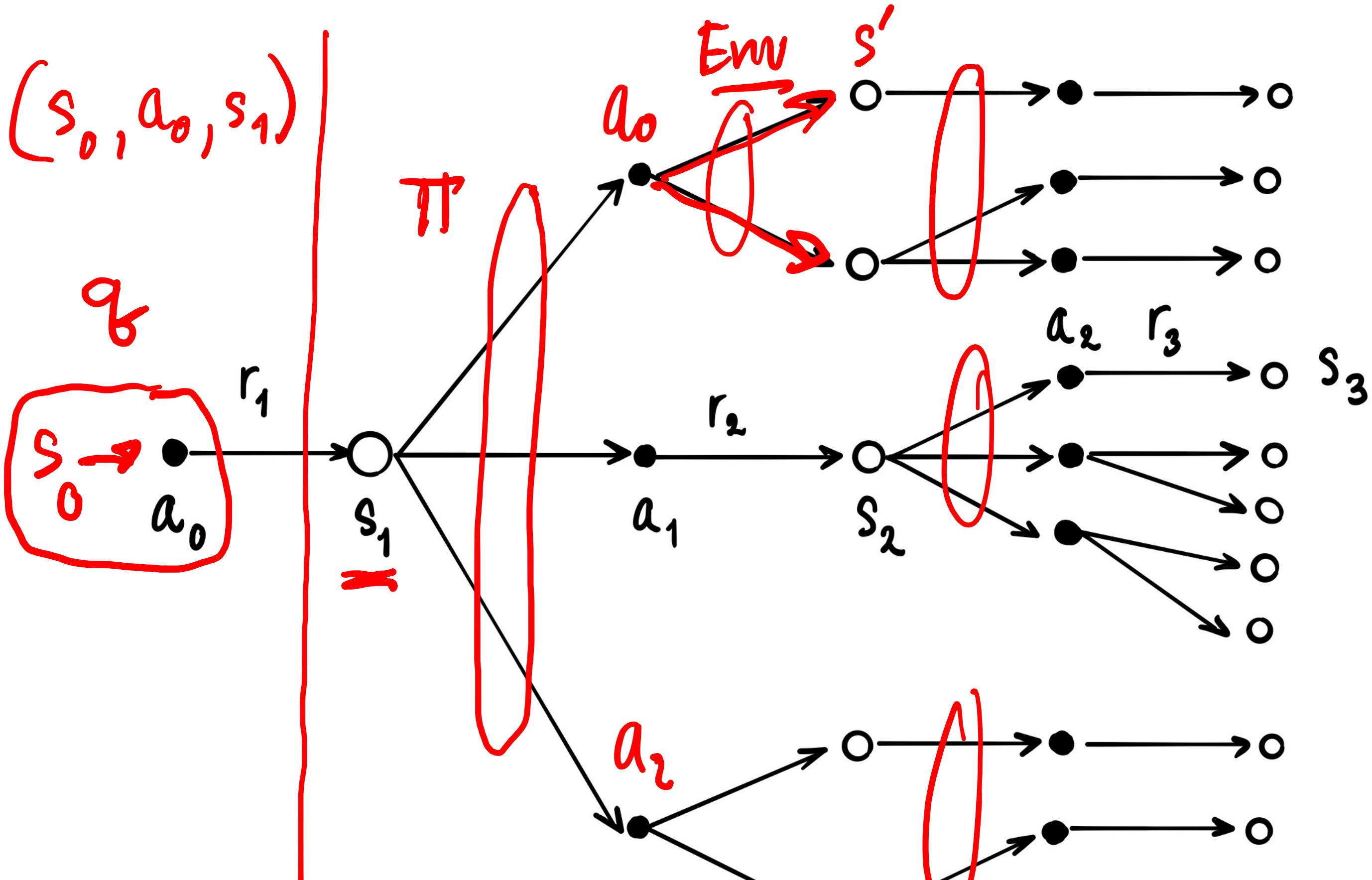
$$q(s_t, a_t) \leftarrow$$

$$q(s_t, a_t) + \alpha\left[r(s_t, a_t) + q(s_{t+1}, \pi(s_{t+1})) - q(s_t, a_t)\right]$$

- We use the "known" next action to correct for behavior policy
- Does this apply to n-step case?

Why it won't work with n-step

- We don't know what are the next-next actions?



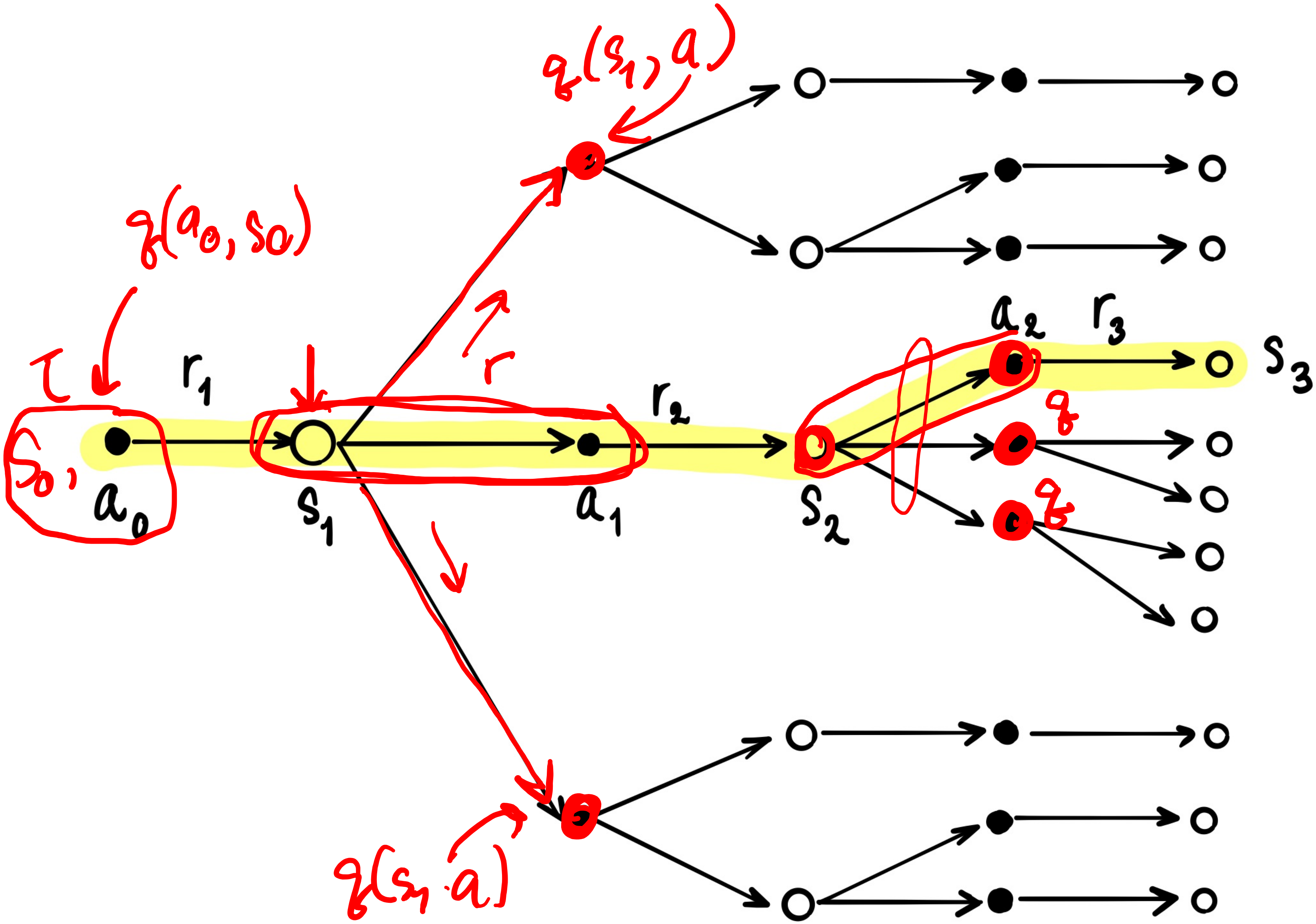
Why it won't work with n-step

- We don't know what are the next-next actions?
  - That needs model
- We could do away with this problem by "bootstrap" them all!
- Tree-backup algorithm (2000)

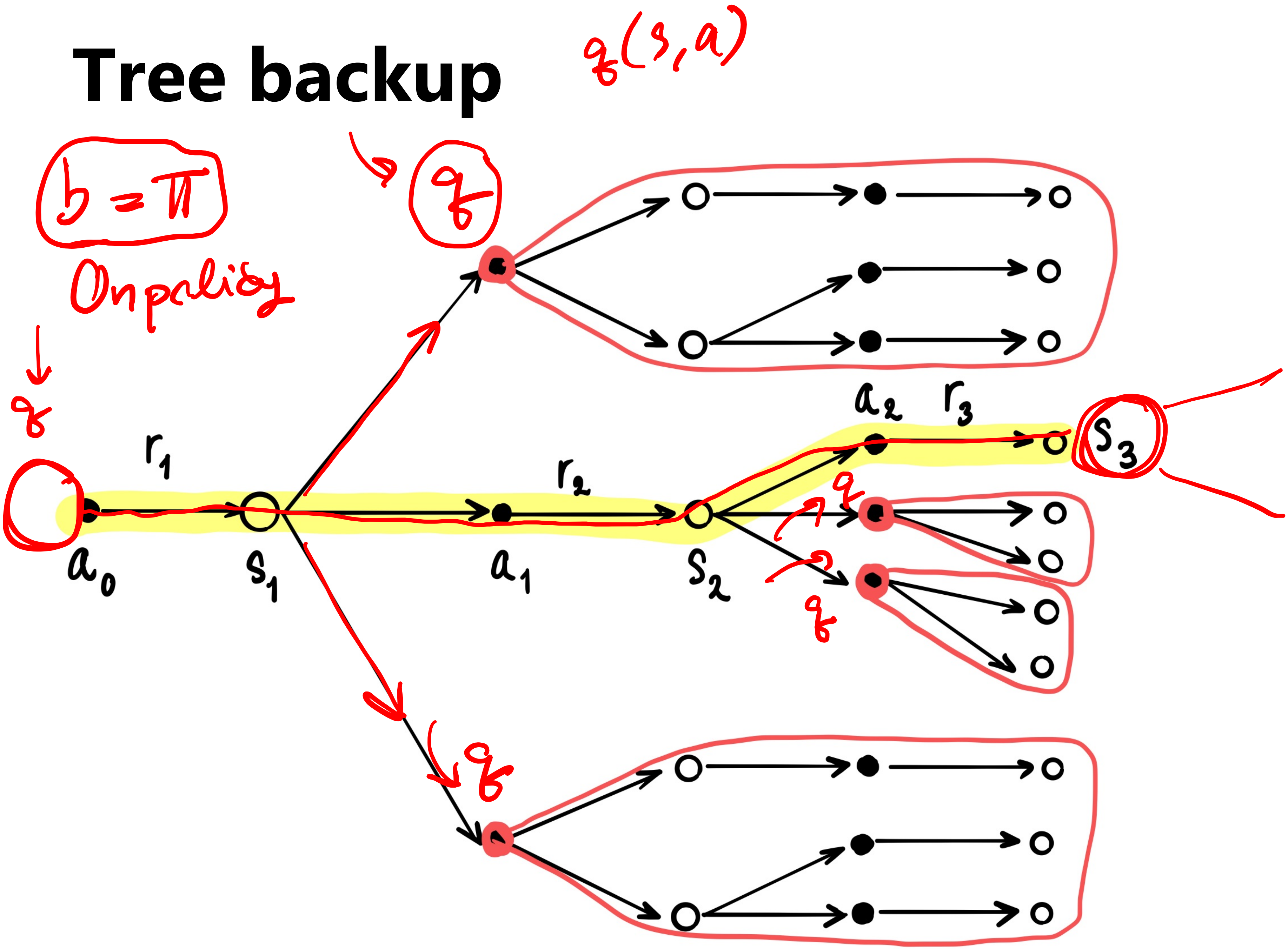
Tree backup

- **N-step off-policy learning without importance sampling**
- We want a kind of “n-step” Expected SARSA
- There is a lot we don’t know
- We bootstrap them all!

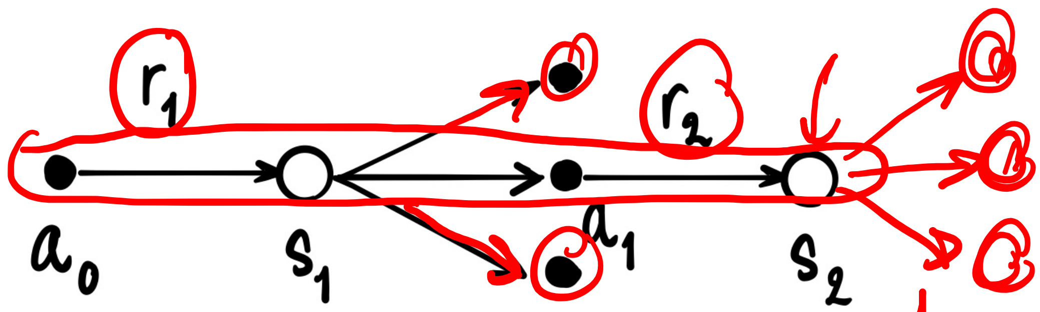
Tree backup



Tree backup



2-step Tree backup algorithm



$$
\begin{aligned}
g_{\text{tb}} = r_{t+1} + \gamma \pi(a_{t+1}|s_{t+1}) &\left[r_{t+2} + \gamma \sum_{a} \pi(a|s_{t+2}) q(s_{t+2}, a) \right] \\
&+ \gamma \sum_{a \neq a_{t+1}} \pi(a|s_{t+1}) q(s_{t+1}, a)
\end{aligned}
$$

$$
q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha [g_{\text{tb}} - q(s_t, a_t)]
$$

Tree backup properties

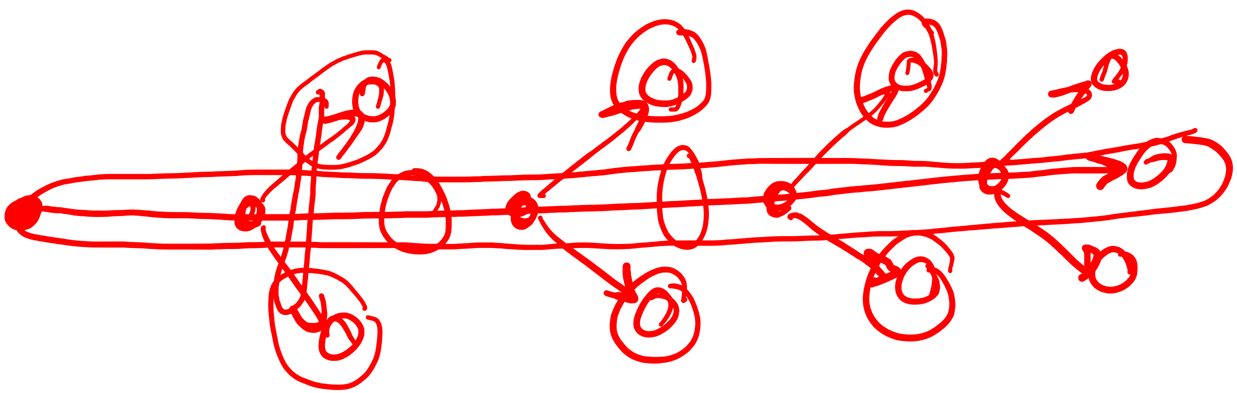
$\tau = (s, a, r, s, a, r)$

- Low variance
- High bias
  - Because we bootstrap almost all the rest
- Bias is not reduced even on-policy
- Doesn't need to know behavior policy

IS

↓

$b(a|s)$ ✗



A bird eye view of on/off-policy

Algorithm	V/Q value	Make it off-policy	Variance	Bias
Monte Carlo	V	IS	High	Low
	Q	IS	High	Low
One-step SARSA	V	IS	Lower	High
	Q	IS	Lower	High
One-step Expected SARSA	V	IS	Lower	High
	Q	Already	Low	High
One-step TD with Deterministic Policy (including Q-learning)	V	IS	Lower	High
	Q	Already	Low	High
N-step SARSA (including lambda)	V	IS	Medium	Medium
	Q	IS	Medium	Medium
Tree backup	V	Already	Low	High
	Q	Already	Low	High