535514: Reinforcement Learning
Lecture 9 — Variance Reduction

Ping-Chun Hsieh

March 21, 2024

Review: Solutions to Variance Reduction

(S1) Baseline ($\equiv$ Set a reference level)

(S2) Critic ($\equiv$ Learn $Q(s, a)$)

(S3) Baseline + Critic ($\equiv$ Advantage function)

Review: Reducing Variance Using a Baseline

- Subtract a baseline $B(s)$ from PG expression of (P2)

$$\mathbb{E}_{\tau \sim P_{\mu}^{\pi_\theta}}\left[\sum_{t=0}^{\infty} \gamma^t \left(Q^{\pi_\theta}(s_t, a_t) - B(s_t) \right) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right]$$

- Subtract a baseline $B(s)$ from PG expression of (P3)

$$\mathbb{E}_{s \sim d_{\mu}^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot \mid s)}\left[\left(Q^{\pi_\theta}(s, a) - B(s) \right) \nabla_\theta \log \pi_\theta(a \mid s) \right]$$

Review: REINFORCE with Baseline

- REINFORCE with baseline

Step 1: Initialize θ_0 and step size η

Step 2: Sample a trajectory $\tau \sim P_\mu^{\pi_\theta}$ and make the update as

$$\theta_{k+1} = \theta_k + \eta\Big(\sum_{t=0}^{\infty} \gamma^t \big(G_t - B(s_t)\big) \nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big)$$

(Repeat Step 2 until termination)

Next Question: *How to choose a good baseline* $B(s)$*?*

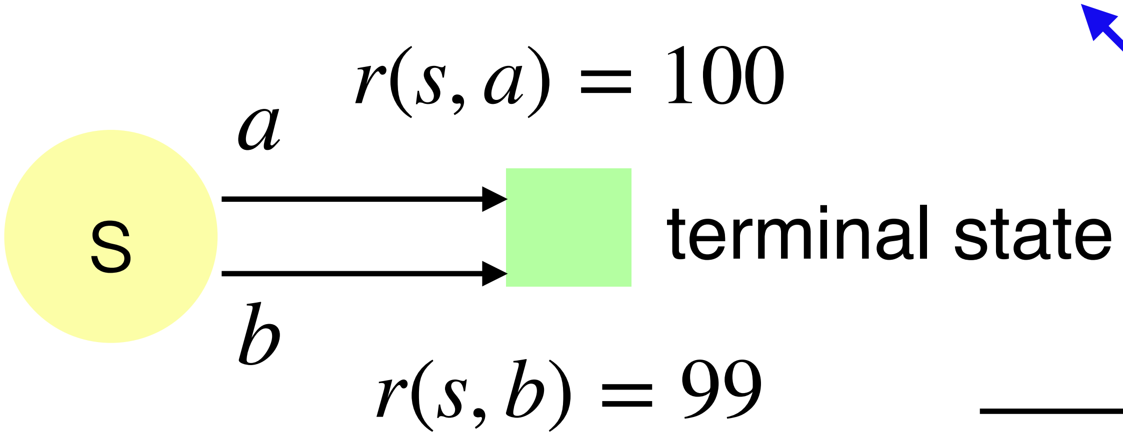
How to Choose the Baseline $B(s)$?

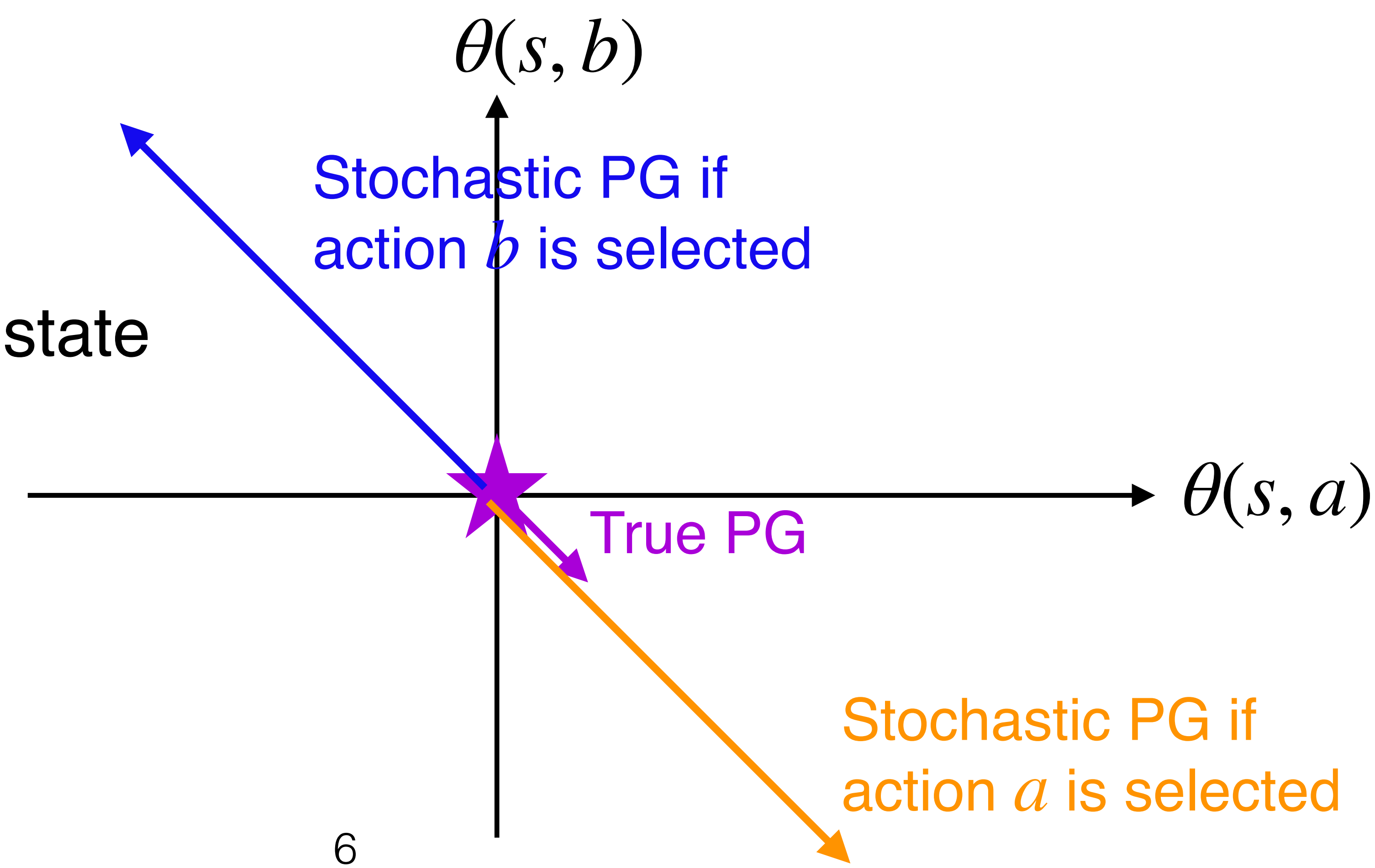
- In REINFORCE, estimate policy gradient with G_t

Original: $$\nabla_\theta V^{\pi_\theta}(\mu) \approx \sum_{t=0}^{\infty} \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$$

With baseline: $$\nabla_\theta V^{\pi_\theta}(\mu) \approx \sum_{t=0}^{\infty} \gamma^t (G_t - B(s_t)) \nabla_\theta \log \pi_\theta(a_t \mid s_t)$$

Let's get some intuition by considering the 1-state MDP example:





Note: In the 1-state MDP example, the trajectory length is 1

The variance of original PG can be written as:

$$\mathbb{V}\Big[G_0 \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_0 \,|\, s_0)\Big] \qquad \text{(Original)}$$

$$= \sum_s P(s_0 = s)\Big(\mathbb{E}[G_0^2(\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_0 \,|\, s))^2 \,|\, s]\Big) - \Big(\sum_s P(s_0 = s)\mathbb{E}[G_0 \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_0 \,|\, s_0) \,|\, s]\Big)^2$$

$$= \sum_s P(s_0 = s)\Big(\sum_a \pi_\theta(a \,|\, s)\mathbb{E}[G_0^2(\frac{\partial}{\partial \theta_i} \log \pi_\theta(a \,|\, s))^2 \,|\, s, a]\Big) - \Big(\sum_s P(s_0 = s)\sum_a \pi_\theta(a \,|\, s)\mathbb{E}[G_t \frac{\partial}{\partial \theta_i} \log \pi_\theta(a \,|\, s) \,|\, s, a]\Big)^2$$

$$= \sum_s P(s_0 = s)\Big(\sum_a \pi_\theta(a \,|\, s)\Big(\frac{\partial}{\partial \theta_i} \log \pi_\theta(a \,|\, s)\Big)^2 \mathbb{E}[G_0^2 \,|\, s, a]\Big) - \Big(\sum_s P(s_0 = s)\sum_a \pi_\theta(a \,|\, s)\frac{\partial}{\partial \theta_i} \log \pi_\theta(a \,|\, s)\mathbb{E}[G_0 \,|\, s, a]\Big)^2$$

The variance of REINFORCE PG with baseline can be written as:

(With baseline)

$$\mathbb{V}\left[(G_0 - B(s_0))\frac{\partial}{\partial\theta_i}\log\pi_\theta(a_0 \mid s_0)\right]$$

$$= \sum_s P(s_0 = s)\Big(\mathbb{E}[(G_0 - B(s))^2(\frac{\partial}{\partial\theta_i}\log\pi_\theta(a_0 \mid s))^2 \mid s]\Big)$$

$$-\Big(\sum_s P(s_0 = s)\mathbb{E}[(G_0 - B(s))\frac{\partial}{\partial\theta_i}\log\pi_\theta(a_0 \mid s_0) \mid s]\Big)^2$$

$$= \sum_s P(s_0 = s)\Big(\sum_a \pi_\theta(a \mid s)(\mathbb{E}[(G_0 - B(s))^2(\frac{\partial}{\partial\theta_i}\log\pi_\theta(a \mid s))^2 \mid s, a]\Big)$$

$$-\Big(\sum_s P(s_0 = s)\sum_a \pi_\theta(a \mid s)\mathbb{E}[(G_0 - B(s))\frac{\partial}{\partial\theta_i}\log\pi_\theta(a \mid s) \mid s, a]\Big)^2$$

$$= \sum_s P(s_0 = s)\Big(\sum_a \pi_\theta(a \mid s)\big(\frac{\partial}{\partial\theta_i}\log\pi_\theta(a \mid s)\big)^2\mathbb{E}[(G_0 - B(s))^2 \mid s, a]\Big)$$

$$-\Big(\sum_s P(s_0 = s)\sum_a \pi_\theta(a \mid s)\frac{\partial}{\partial\theta_i}\log\pi_\theta(a \mid s)\mathbb{E}[G_0 \mid s, a]\Big)^2$$

Quantifying Variance Reduction By $B(s)$

$$\mathbb{V}\Big[G_0 \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_0 \,|\, s_0)\Big] - \mathbb{V}\Big[(G_0 - B(s_0)) \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_0 \,|\, s_0)\Big]$$

$$= \sum_s P(s_0 = s)$$

$$\Bigg(\sum_a \underbrace{\pi_\theta(a \,|\, s) \Big(\frac{\partial}{\partial \theta_i} \log \pi_\theta(a \,|\, s) \Big)^2}_{:= c_a} \big(\mathbb{E}[G_0^2 \,|\, s, a] - \mathbb{E}[(G_0 - B(s))^2 \,|\, s, a] \big) \Bigg)$$

$$= \sum_s P(s_0 = s) \sum_a c_a \big(\mathbb{E}[2B(s)G_0 - B(s)^2 \,|\, s, a] \big)$$

- Suppose $\mathbb{E}[G_0 \,|\, s, a] \equiv Q^{\pi_\theta}(s, a) \approx V^{\pi_\theta}(s)$, then we may choose $B(s) = V^{\pi_\theta}(s)$
- In practice, $B(s) = V^{\pi_\theta}(s)$ is a popular choice

(S2) Reducing Variance Using a Critic

- Monte Carlo policy gradient requires G_t, which has high variance
- Recall:

  (P3) Q-value and discounted state visitation:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[Q^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a\,|\,s)\Big]$$

- Idea: Learn a critic to estimate action-value function

$$Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$$

(S2) Reducing Variance Using a Critic (Cont.)

- Actor-critic algorithms maintain 2 sets of parameters
  - Critic: updates action-value function parameter w
  - Actor: updates policy parameters θ, in the direction suggested by critic

- Actor-critic algorithms follow an approximate policy gradient

$$\nabla_\theta V^{\pi_\theta}(\mu) \approx \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[Q_w(s,a) \nabla_\theta \log \pi_\theta(a \mid s) \right]$$

- Stochastic PG methods would use the following for policy update

$$Q_w(s,a) \nabla_\theta \log \pi_\theta(a \mid s)$$

Q-Value Actor-Critic Algorithm

- A simple actor-critic algorithm based on a Q-function critic

Step 1: Initialize θ, w, step size η, s_0 and sample $a_0 \sim \pi_\theta$

Step 2: For each step $t = 0,1,2,\cdots$

Sample reward r_{t+1}; sample transition s_{t+1}

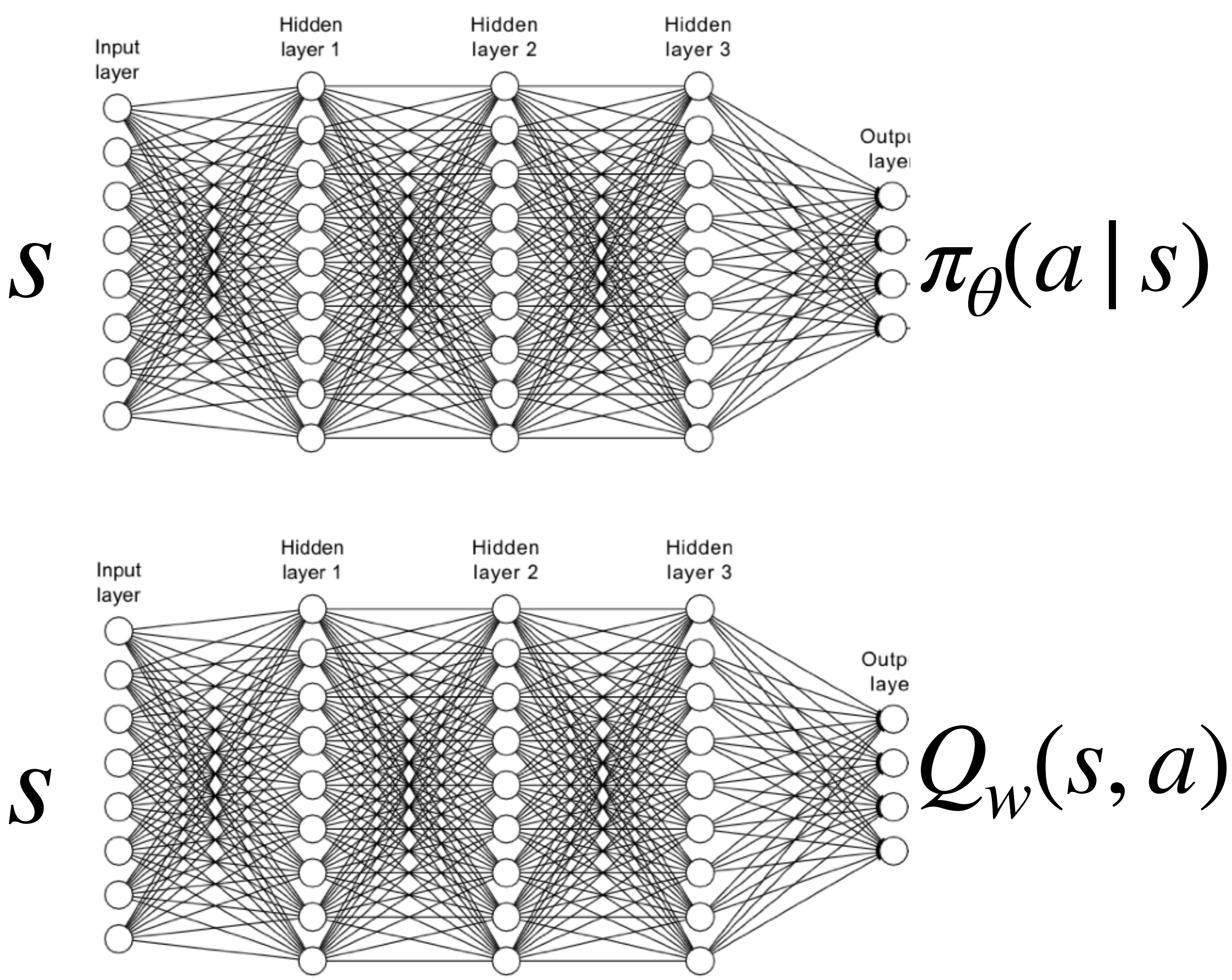
Sample action $a_{t+1} \sim \pi_\theta(s_{t+1}, a_{t+1})$

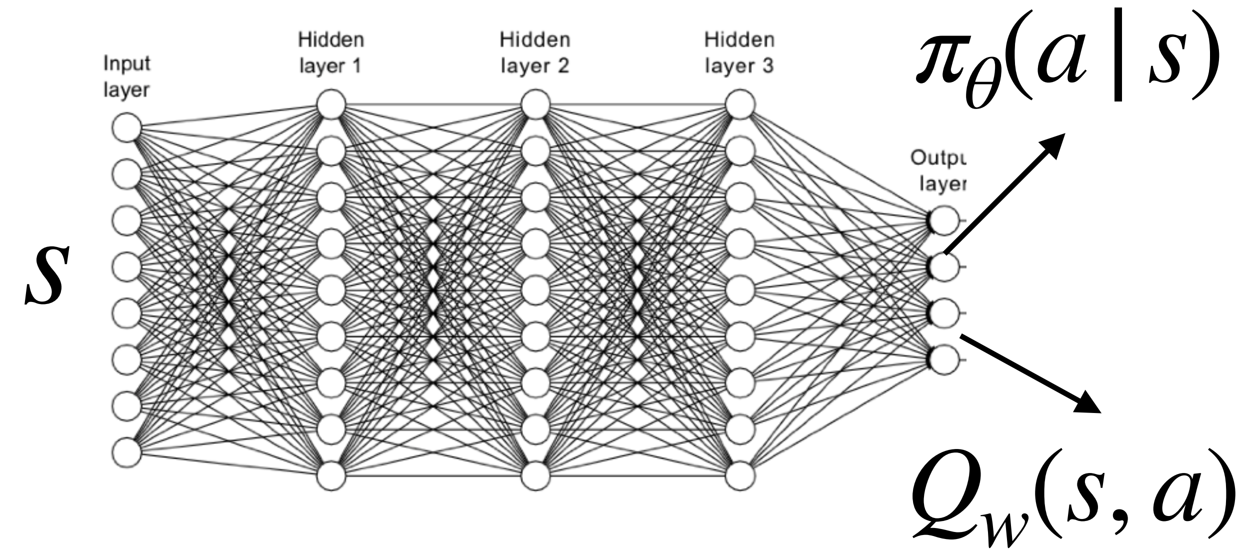
$$\theta \leftarrow \theta + \eta Q_w(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \,|\, s_t)$$

Update w for $Q_w(s, a)$ (possibly using $r_{t+1}, s_{t+1}, a_{t+1}$)

Actor-Critic Architecture

- Two popular choices:





Two separate networks

One shared network

(S3) Reducing Variance Using Advantage Functions

- Question: Can we combine both baseline and critic?

- Define advantage function as

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$

- Recall:

  (P3) Q-value and discounted state visitation:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[Q^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a\,|\,s)\Big]$$

- We have:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a\,|\,s)\Big]$$

Policy Gradient With Advantage Functions

- **Policy Gradient With Advantage Function**:

(P4) Advantage:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[A^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a\,|\,s)\right]$$

(P5) REINFORCE with advantage:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau\sim P_\mu^{\pi_\theta}}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi_\theta}(s_t,a_t)\,\nabla_\theta \log \pi_\theta(a_t\,|\,s_t)\right]$$

Optimal Baseline for Variance Reduction?

The Optimal Reward Baseline for Gradient-Based Reinforcement Learning

Lex Weaver
Department of Computer Science
Australian National University
ACT AUSTRALIA 0200
Lex.Weaver@cs.anu.edu.au

Nigel Tao
Department of Computer Science
Australian National University
ACT AUSTRALIA 0200
Nigel.Tao@cs.anu.edu.au

Abstract

There exist a number of reinforcement learning algorithms which learn by climbing the gradient of expected reward. Their long-run convergence has been proved, even in partially observable environments with non-deterministic actions, and without the need for a system model. However, the variance of the gradient estimator has been found to be a significant practical problem. Recent approaches have discounted future rewards, introducing a bias-variance trade-off into the gradient estimate. We incorporate a reward baseline into the learning system, and show that it affects variance without introducing further bias. In particular, as we approach the zero-bias, high-variance parameterization, the optimal (or variance minimizing) constant reward baseline is equal to the long-term average expected reward. Modified policy-gradient algorithms are presented, and a number of experiments demonstrate their improvement over previous work.

the reliance on both a system model and a need to identify a specific recurrent state, and operate in partially observable environments with non-deterministic actions (POMDPs).

However, the variance of the gradient estimator remains a significant practical problem for policy-gradient applications, although discounting is an effective technique. Discounting future rewards introduces a bias-variance trade-off: variance in the gradient estimates can be reduced by heavily discounting future rewards, but the estimates will be biased; the bias can be reduced by not discounting so heavily, but the variance will be higher. Our work complements the discounting technique by introducing a *reward baseline*[1] which is designed to reduce variance, especially as we approach the zero-bias, high-variance discount factor.

The use of a reward baseline has been considered a number of times before, but we are not aware of any analysis of its effect on variance in the context of the recent policy-gradient algorithms. (Sutton, 1984) empirically investigated the inclusion of a reinforcement comparison term in several stochastic learning equations, and argued that it should result in faster learning for unbalanced reinforce-

- One could find an optimal baseline $b^*(s)$ by directly minimizing the covariance of a PG estimator

(A practice problem of HW2)

[UAI 2001]
Available at https://arxiv.org/pdf/1301.2315.pdf

How to Estimate the Action-Value Function?

- A critic = solving the policy evaluation problem
  - How good is a policy π_θ?

- In Lecture 3, we discussed both non-iterative and iterative policy evaluation given the MDP model parameters

- Question: How to do policy evaluation without knowing MDP model parameters?

Next Topic: Model-free prediction!

Model-Free Prediction

= Policy Evaluation with **Unknown** Dynamics & Rewards

3 Major Approaches for Model-Free Prediction

1. Monte Carlo (MC)

2. Temporal Difference: TD(0) and n-step TD

3. TD(λ) and GAE

References:

Richard Sutton and Andrew Barto, Reinforcement Learning: An Introduction, 2019

Singh and Sutton, "Reinforcement Learning with Replacing Eligibility Traces," ML 1996

Schulman et al., High-Dimensional Continuous Control Using Generalized Advantage Estimation, ICLR 2016

Monte-Carlo for Policy Evaluation

- Recall: Monte-Carlo policy gradient
  - Use sample return G_t for the estimate of policy gradient

$$\nabla_\theta V^{\pi_\theta}(\mu) \approx \sum_{t=0}^{\infty} \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$$

- Question: Can we use the same idea for policy evaluation (i.e., finding $V^\pi(s)$)?

Monte-Carlo for Policy Evaluation (Cont.)

To find the value function V^π under a fixed policy π:

- For episodic environments ➡ sample a set of trajectories $\{\tau^{(i)}\}_{i=1}^{K}$ and calculate average returns $\frac{1}{K}\sum_{i=1}^{K} G(\tau^{(i)}) \approx V^\pi(\mu)$

- For continuing environments ➡ sample a set of trajectories (but with proper ***truncation***) and calculate average returns as an estimate of $V^\pi(\mu)$

Features of MC

1. MC is model-free
    - MC learns directly from episodes *without* estimating MDP transition probabilities or reward function

2. MC learns from complete episodes

Is MC Policy Evaluation Useful in Practice?

Yes! MC serves as a pseudo-oracle for true $V^\pi(s)$ or $Q^\pi(s, a)$

Example: Finding the "true value functions" in the TD3 paper

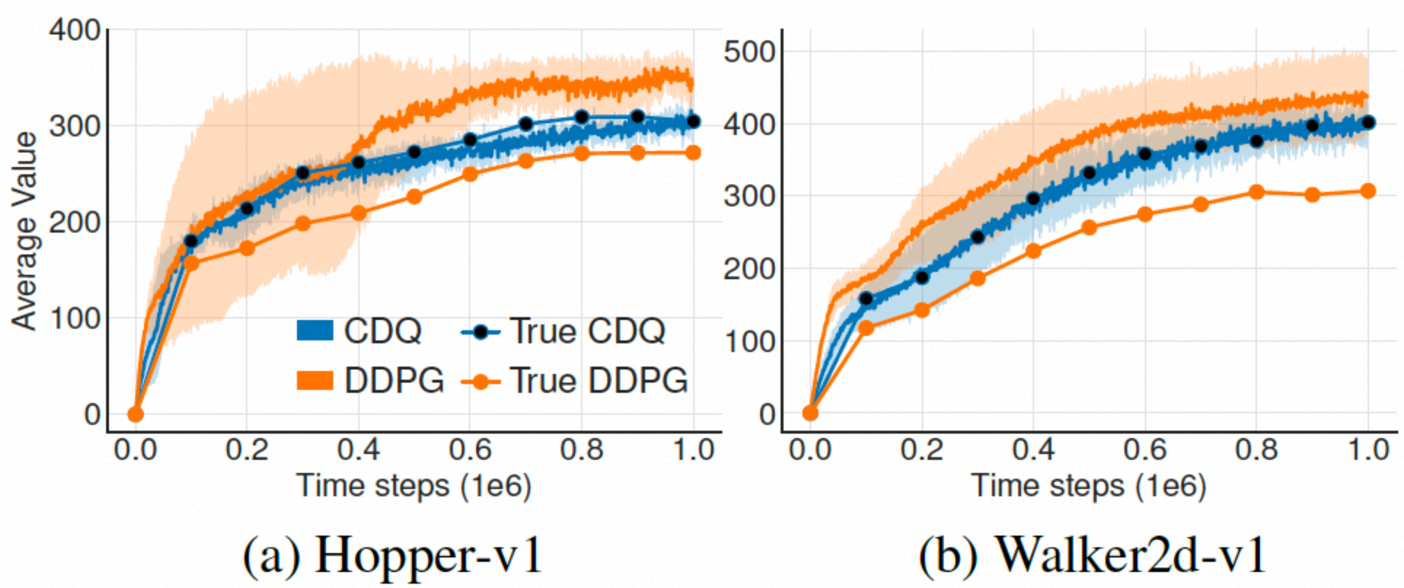
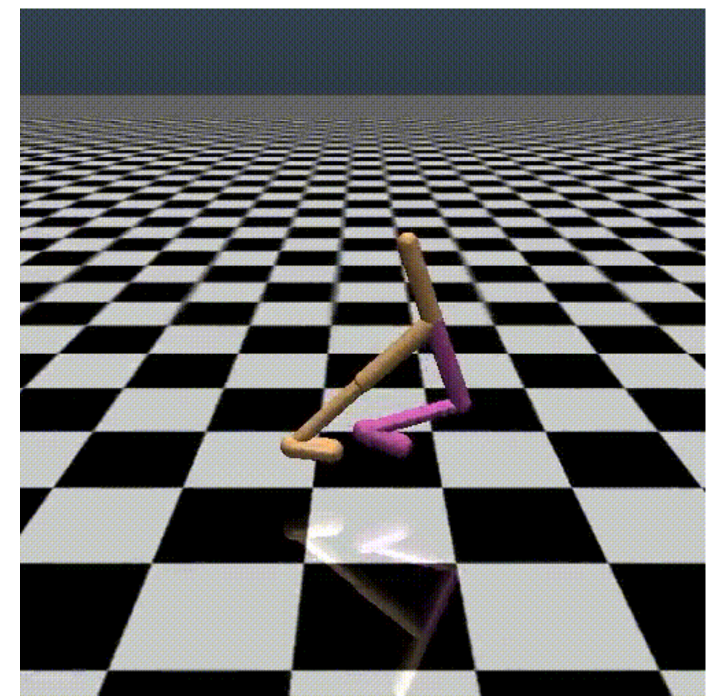


Figure 1. Measuring overestimation bias in the value estimates of DDPG and our proposed method, Clipped Double Q-learning (CDQ), on MuJoCo environments over 1 million time steps.

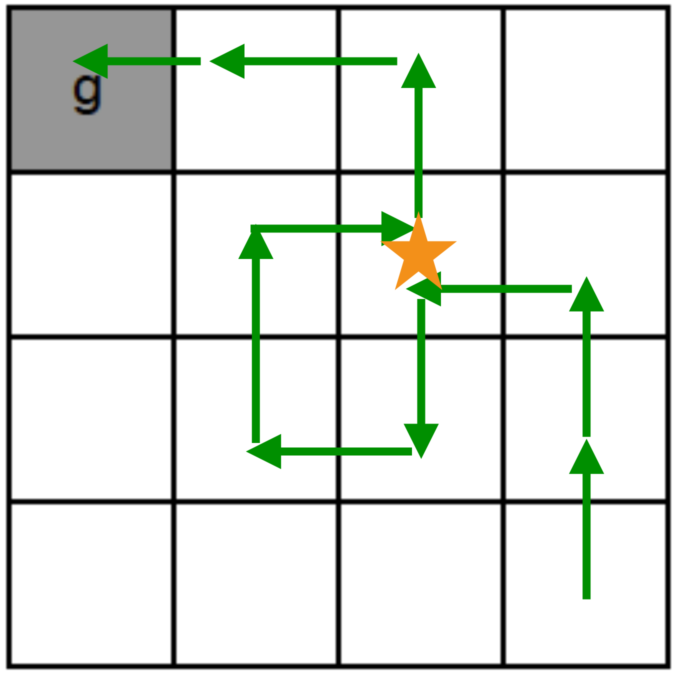


- If the policy is *deterministic*, how many trajectories do we need?
- What if the policy is *stochastic*?

Fujimoto et al., Addressing Function Approximation Error in Actor-Critic Methods, ICML 2018

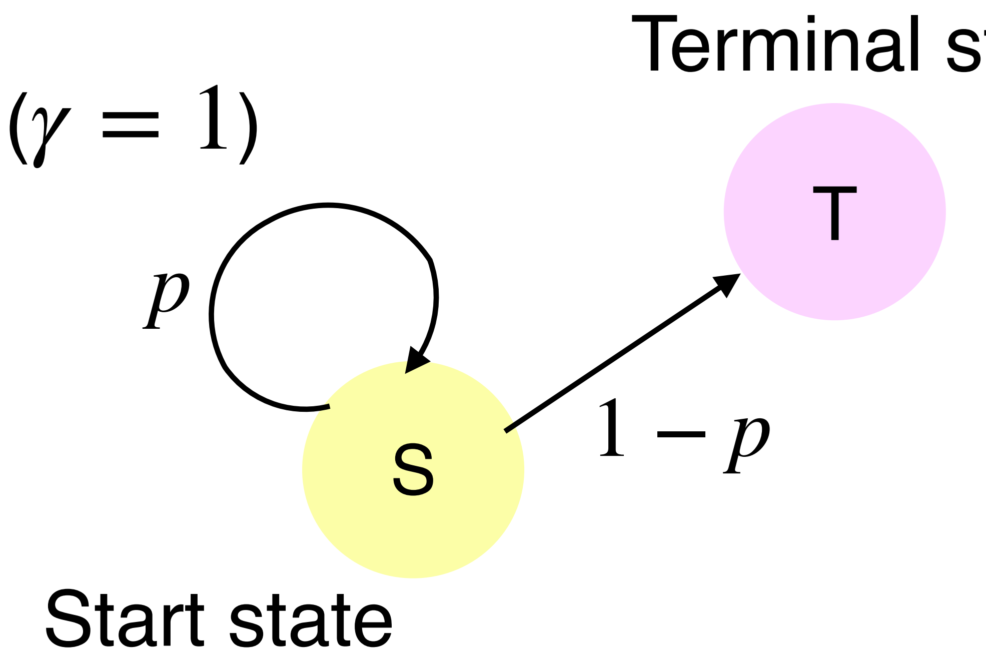
Two Variants of MC Policy Evaluation: First-Visit and Every-Visit

- A visit to s: an occurrence of a state s in an episode
- First-visit MC: Estimate the value of a state as the average of the returns that have followed the first visit to the state in an episode
- Every-visit MC: Estimate the value of a state as the average of the returns that have followed all visits to the state



Example: First visit to ★? How many visits to ★?

Example: 2-State MRP



@Start state: reward = 1

@Terminal state: reward = 0

- Consider a sample trajectory: $S \rightarrow S \rightarrow S \rightarrow S \rightarrow T$
- Question: First-visit MC estimate of $V(S)$ = ? 4
- Question: Every-visit MC estimate of $V(S)$ = ?

$$(4 + 3 + 2 + 1)/4 = 2.5$$

- Question: Which estimate is better?

First-Visit MC Policy Evaluation (Formally)

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

(Slide Credit: Emma Brunskill)

Every-Visit MC Policy Evaluation (Formally)

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For **every** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

(Slide Credit: Emma Brunskill)

An Incremental Expression of Sample Mean

- Let $z_1, z_2, z_3 \cdots$ be a sequence of real numbers
- Sample mean of $z_1, \cdots, z_n$ is denoted by $\bar{z}_n$

$$\bar{z}_n := \frac{1}{n}\sum_{k=1}^{n} z_k = \frac{1}{n}\left(z_n + \sum_{k=1}^{n-1} z_k\right)$$

$$= \frac{1}{n}\left(z_n + (n-1)\bar{z}_{n-1}\right)$$

$$= \frac{1}{n}\left(z_n + (n-1)\bar{z}_{n-1} + \bar{z}_{n-1} - \bar{z}_{n-1}\right)$$

$$= \bar{z}_{n-1} + \frac{1}{n}\left(z_n - \bar{z}_{n-1}\right)$$

Incremental Monte-Carlo Updates

(Alternative expression of every-visit MC)

- Update $V^{\pi}(s)$ incrementally after each episode

  $s_0, a_0, r_1, \cdots, s_{T-1}, a_{T-1}, r_T$

  For each state s_t with sample return G_t

$$N(s_t) \leftarrow N(s_t) + 1$$
$$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G_t - V(s_t))$$

- In non-stationary environments, we may instead track the exponential moving average (i.e. forget old episodes) by

$$V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t))$$