

535514: Reinforcement Learning

Lecture 6 — Policy Gradient

Ping-Chun Hsieh

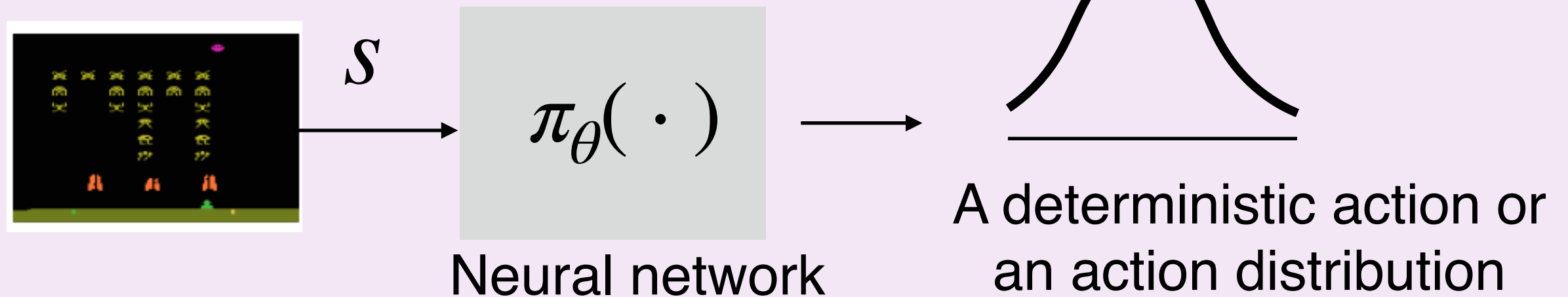
March 11, 2024

Policy Optimization Framework:

- ▶ Consider a parametric policy class: $\{\pi_\theta \mid \theta \in \Theta \subset \mathbb{R}^d\}$
- ▶ View RL as an optimization problem: $\max_{\theta} V^{\pi_\theta}(\mu) := \mathbb{E}_{s_0 \sim \mu}[V^{\pi_\theta}(s_0)]$

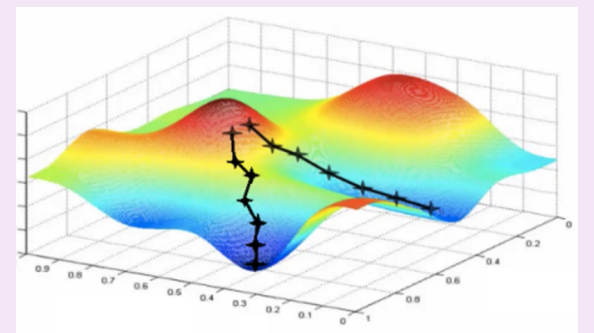
Example:

1. Parametrize the policy by a neural network



2. Update π_θ iteratively, e.g., by gradient ascent (GA)

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} V^{\pi_\theta}(\mu)$$



- ▶ **Question:** Why using $V^{\pi}(\mu)$ as the objective (instead of each individual $V^{\pi}(s)$)?

3 Challenges in Policy Optimization

(C1) How to parametrize the policy?

(C2) How to solve the optimization problem in a model-free manner?

1. Policy gradient (PG)

2. Model-free prediction (aka model-free policy evaluation)

(C3) Any theoretical guarantee of PG?

Typical Examples of Parametric Policies

Discrete action space

- ▶ Softmax policies:

$$\pi_{\theta}(a | s) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}$$

- ▶ Linear softmax policies:

$$\pi_{\theta}(a | s) = \frac{\exp(\theta^T \phi_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta^T \phi_{s,a'})}$$

- ▶ Neural softmax policies:

$$\pi_{\theta}(a | s) = \frac{\exp(f_{\theta}(s, a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\theta}(s, a'))}$$

Continuous action space

- ▶ Gaussian policies:

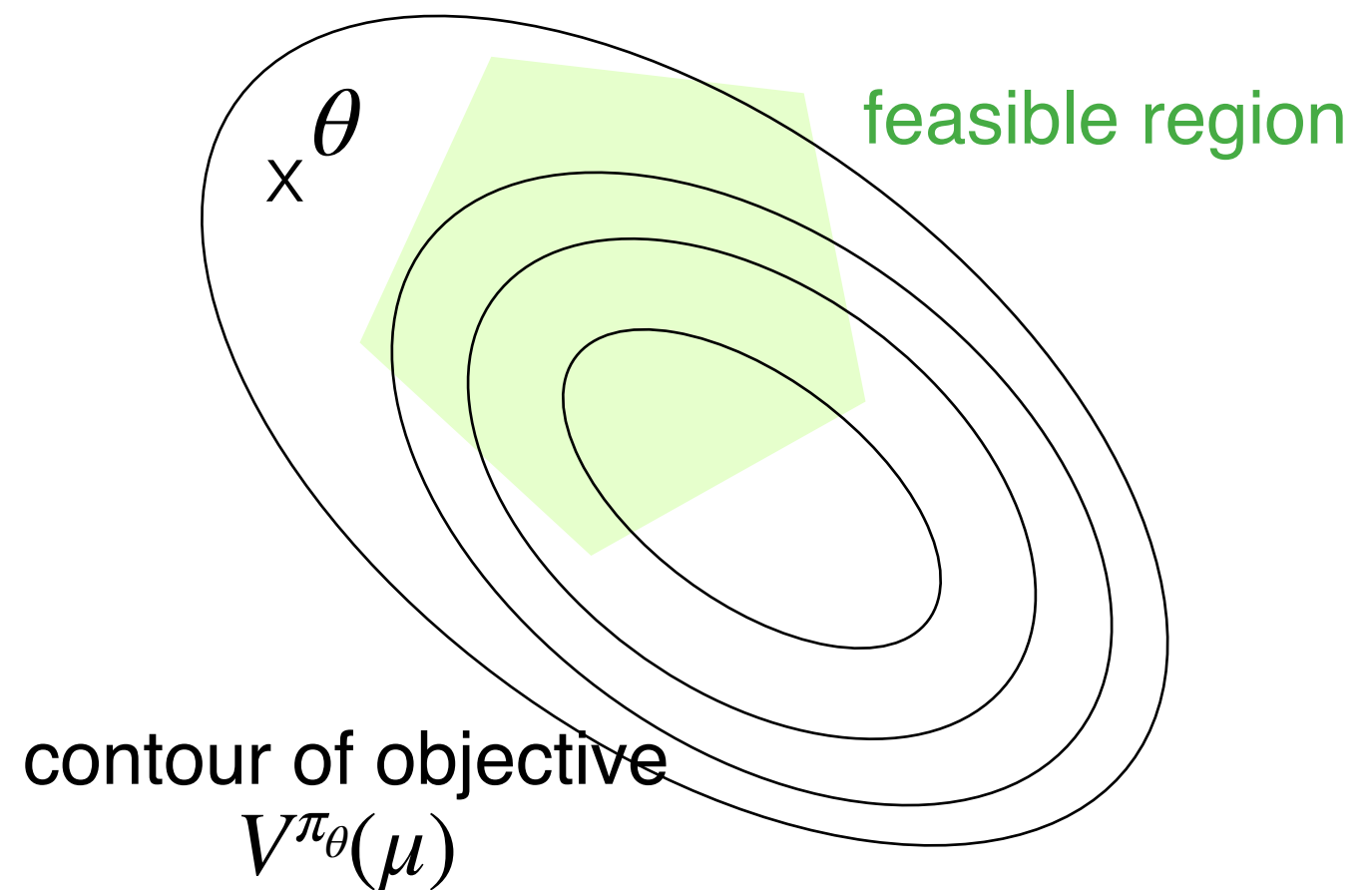
$$a \sim \mathcal{N}(f_{\theta}(s), \sigma^2)$$

Questions: Are these parametric policies general enough?

How About Direct Tabular Parametrization?

- ▶ **Question**: Can we do direct parametrization, i.e., $\pi_{\theta}(a | s) = \theta_{s,a}$?
- ▶ Possible, but now we face **constrained** policy optimization

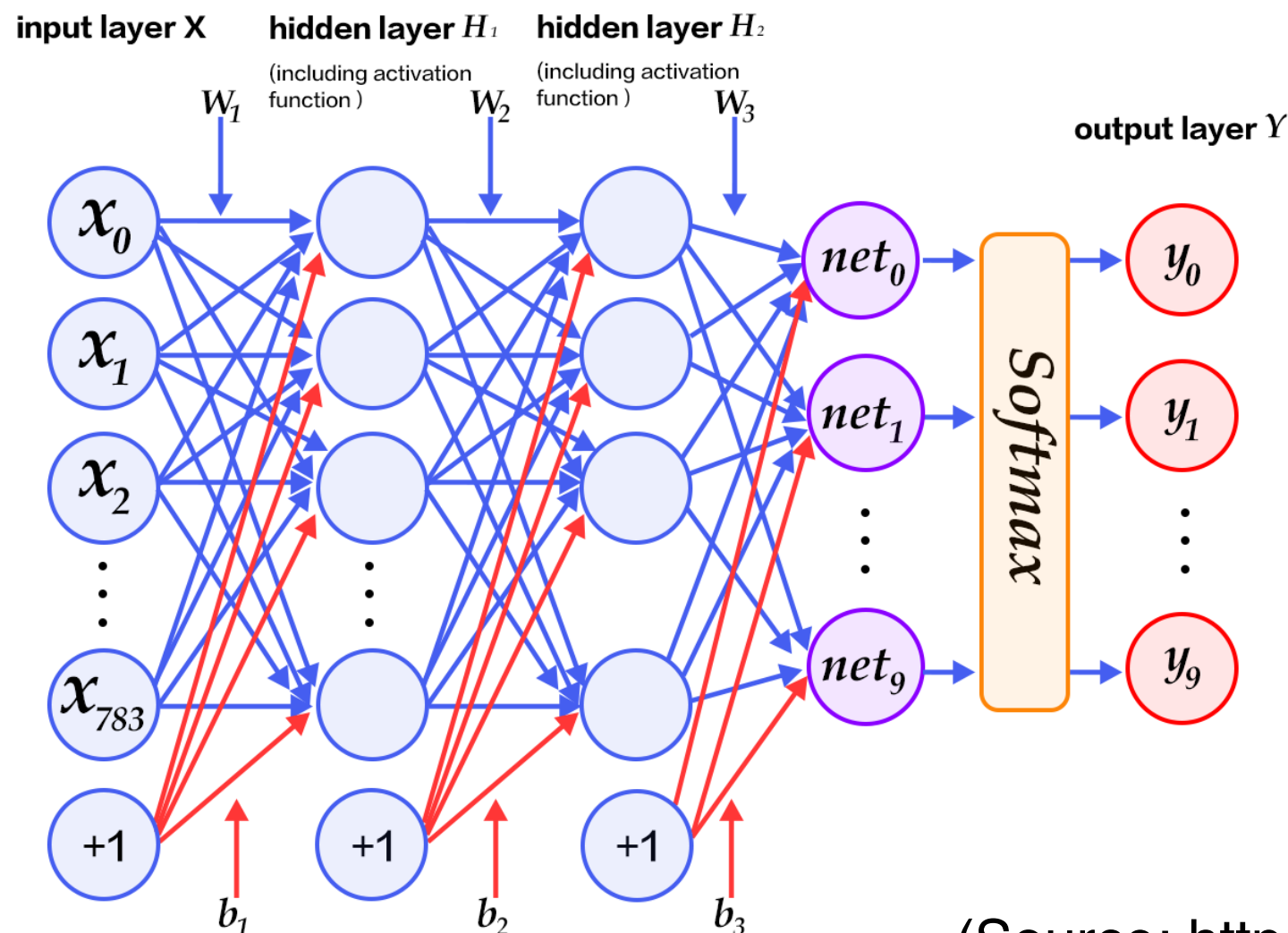
$$\begin{aligned} & \max_{\theta \in \Theta} V^{\pi_{\theta}}(\mu) \\ & \text{subject to } \sum_a \theta_{s,a} = 1, \forall s \end{aligned}$$



- ▶ In this case, **projection** is needed to ensure a valid probability distribution (often computationally heavy)

Neural Softmax Policies

- ▶ Neural softmax policies:
$$\pi_{\theta}(a | s) = \frac{\exp(f_{\theta}(s, a))}{\sum_{a' \in \mathcal{A}} \exp(f_{\theta}(s, a'))}$$
- ▶ θ denotes all the weights and biases of a neural net



(Source: <https://files.ifl.uzh.ch/ddis/>)

Policy Gradient (PG)

Historical Accounts on PG

Policy Gradient Methods for Reinforcement Learning with Function Approximation

Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour
AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932

Abstract

Function approximation is essential to reinforcement learning, but the standard approach of approximating a value function and determining a policy from it has so far proven theoretically intractable. In this paper we explore an alternative approach in which the policy is explicitly represented by its own function approximator, independent of the value function, and is updated according to the gradient of expected reward with respect to the policy parameters. Williams's REINFORCE method and actor-critic methods are examples of this approach. Our main new result is to show that the gradient can be written in a form suitable for estimation from experience aided by an approximate action-value or advantage function. Using this result, we prove for the first time that a version of policy iteration with arbitrary differentiable function approximation is convergent to a locally optimal policy.



Richard Sutton
(U of Alberta & DeepMind)



Satinder Singh
(UMich & DeepMind)

A seminal paper on PG in NIPS 1999

Review: Some Useful Notations

- ▶ **Question:** To apply GA, what do we need?
- ▶ Policy gradient! $\nabla_{\theta} V^{\pi_{\theta}}(\mu)$ (μ : distribution of initial state)

Some useful notations

- ▶ 1. Sample return $G(\tau)$ along a trajectory $\tau = (s_0, a_0, r_1, \dots)$

$$G(\tau) := \sum_{t=0}^{\infty} \gamma^t \cdot r_{t+1}(\tau)$$

- ▶ 2. Value function written in $R(\tau)$

$$V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}}[G(\tau)] = \sum_{\tau} G(\tau) P_{\mu}^{\pi_{\theta}}(\tau)$$

Review: Some Useful Notations (Cont.)

- ▶ 3. Discounted state visitation distribution (aka “occupancy measure”)

$$d_{s_0}^{\pi}(s) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid s_0, \pi)$$

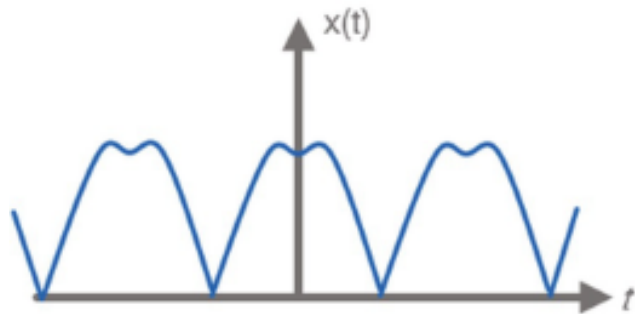
$$d_{\mu}^{\pi}(s) := \mathbb{E}_{s_0 \sim \mu} [d_{s_0}^{\pi}(s)]$$

- ▶ **Question:** What’s the purpose of $(1 - \gamma)$ in the above?

State Visitation Distribution: An Analogy

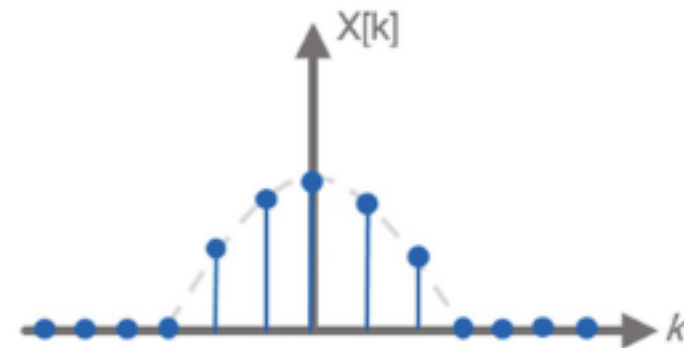
Fourier
Series

Time domain



\equiv

Frequency domain



RL
Objective
Function

Time domain

$$\mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 \sim \mu; \pi_{\theta} \right]$$

$=$

Distribution domain

$$\mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta}}, a \sim \pi(\cdot | s)} [R(s, a)]$$

(We will use this form several times in the sequel)

The Policy Gradient: Inherent Difficulty

► **Issue:** Computing $\nabla_{\theta} V^{\pi_{\theta}}(\mu)$ might seem highly non-trivial...

► Recall: $V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}}[G(\tau)] = \sum_{\tau} G(\tau) P_{\mu}^{\pi_{\theta}}(\tau)$

► $V^{\pi_{\theta}}(\mu)$ is affected by θ in two ways:

1. θ determines $\pi_{\theta}(\cdot | s)$

2. The actions taken by π_{θ} affect the state visitation accordingly

Expressions of The Policy Gradient

► Expressions of Policy Gradient (aka Policy Gradient Theorem):

(P1) Total reward: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

(P2) REINFORCE: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

(P3) Q-value and discounted state visitation:

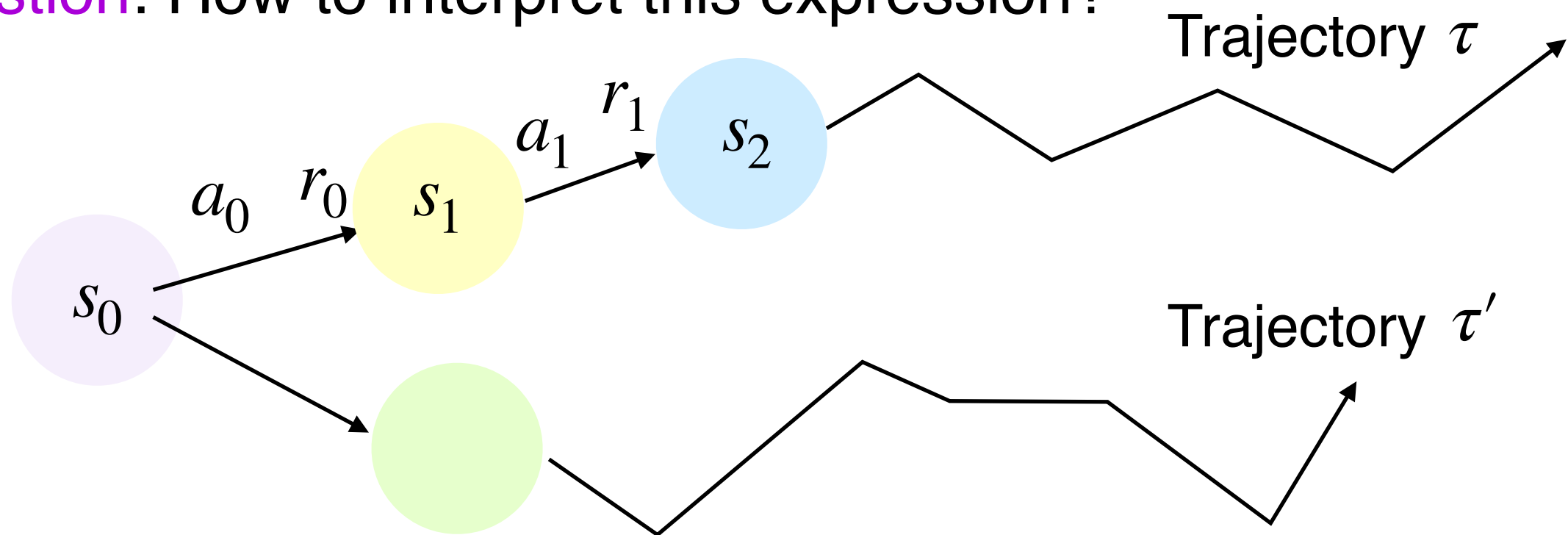
$$\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta}}} \mathbb{E}_{a \sim \pi_{\theta}(\cdot | s)} \left[Q^{\pi_{\theta}}(s, a) \nabla_{\theta} \log \pi_{\theta}(a | s) \right]$$

- Question: Why do we care about various expressions?
- Each expression gives us an RL algorithm!

(P1) Total Reward Policy Gradient

► Want to show: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

► **Question:** How to interpret this expression?



1. $G(\tau) > 0$:

2. $G(\tau) < 0$:

► **Question:** Can you find out any potential issue?

(P1) Total Reward Policy Gradient (Cont.)

► Want to show: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

► Recall: $V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} [G(\tau)] = \sum_{\tau} G(\tau) P_{\mu}^{\pi_{\theta}}(\tau)$

$$\begin{aligned} \nabla_{\theta} V^{\pi_{\theta}}(\mu) &= \sum_{\tau} G(\tau) \nabla_{\theta} P_{\mu}^{\pi_{\theta}}(\tau) \\ &= \sum_{\tau} G(\tau) \left(P_{\mu}^{\pi_{\theta}}(\tau) \cdot \nabla_{\theta} \log P_{\mu}^{\pi_{\theta}}(\tau) \right) \\ &= \sum_{\tau} G(\tau) \left(P_{\mu}^{\pi_{\theta}}(\tau) \cdot \nabla_{\theta} \log (\mu(s_0) \pi_{\theta}(a_0 | s_0) P(s_1 | s_0, a_0) \pi_{\theta}(a_1 | s_1) \cdots) \right) \\ &= \sum_{\tau} G(\tau) \left(P_{\mu}^{\pi_{\theta}}(\tau) \cdot \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi(a_t | s_t) \right) \\ &= \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right] \end{aligned}$$

(Not a rigorous proof)

Technical Subtlety in the Proof of (P1):

- ▶ For τ of infinite length, $\nabla_{\theta} P_{\mu}^{\pi_{\theta}}(\tau)$ needs to be handled more carefully
- ▶ **Idea:** Consider “truncated trajectories” and take the limit

1. Without loss of generality, assume $R(s, a) \in [0, 1]$

Define $\tau^{(k)}$ to be the truncated version of τ up to step k

$$\text{Define } V_k^{\pi_{\theta}}(\mu) = \sum_{\tau^{(k)}} G(\tau^{(k)}) P_{\mu}^{\pi_{\theta}}(\tau^{(k)})$$

2. Then, $\lim_{k \rightarrow \infty} V_k^{\pi_{\theta}}(\mu) = V^{\pi_{\theta}}(\mu)$ (by Monotone convergence theorem)

3. Next, we need to verify that $\lim_{k \rightarrow \infty} \nabla_{\theta} V_k^{\pi_{\theta}}(\mu) = \nabla_{\theta} V^{\pi_{\theta}}(\mu)$ by

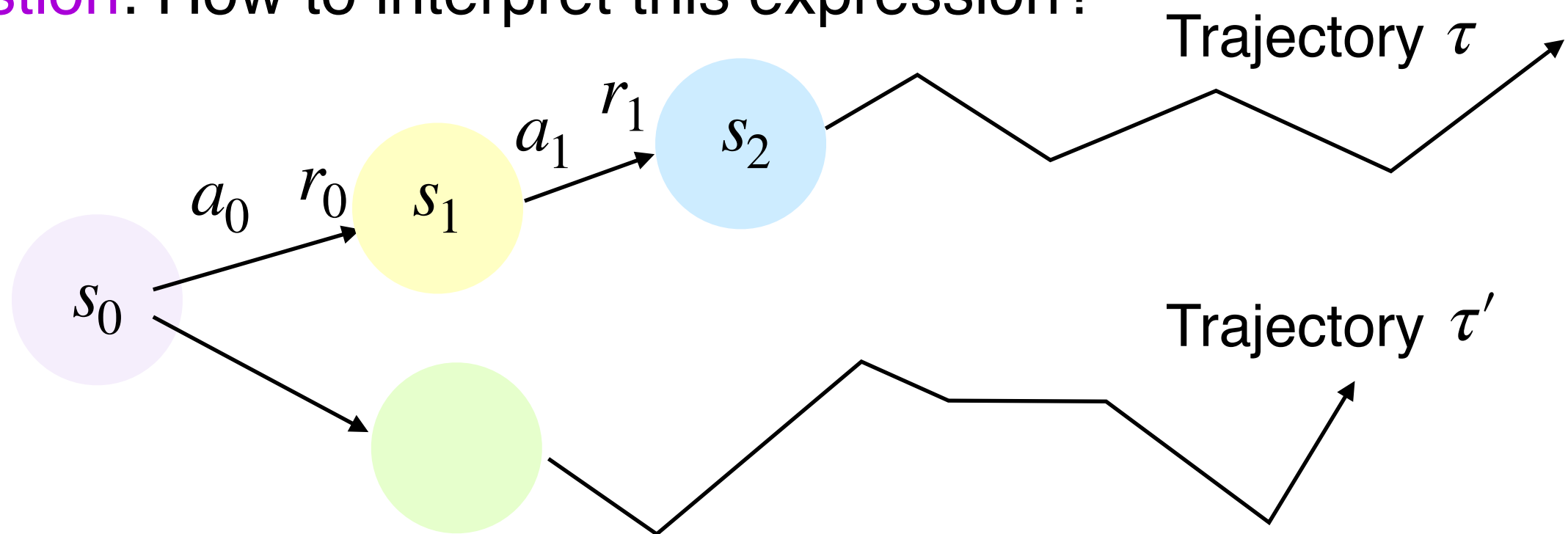
showing “uniform convergence” of $\nabla_{\theta} V_k^{\pi_{\theta}}(\mu)$ [Rudin 1976, Thm 7.17]

(Note that it is not always valid to interchange ∇ and \lim)

(P2) REINFORCE Policy Gradient

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

► **Question:** How to interpret this expression?



1. $Q(s, a) > 0$:

2. $Q(s, a) < 0$:

► **Question:** Can you find out any potential issue?

(P2) REINFORCE Policy Gradient

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

► **Idea:** prove by induction

► Recall: $V^{\pi_{\theta}}(s_0) = \sum_{a_0 \in \mathcal{A}} \pi_{\theta}(a_0 | s_0) Q^{\pi_{\theta}}(s_0, a_0)$

$$\begin{aligned} \nabla_{\theta} V^{\pi_{\theta}}(s_0) &= \sum_{a_0 \in \mathcal{A}} \nabla_{\theta} \left(\pi_{\theta}(a_0 | s_0) Q^{\pi_{\theta}}(s_0, a_0) \right) \\ &= \underbrace{\sum_{a_0} \nabla_{\theta} \left(\pi_{\theta}(a_0 | s_0) \right) Q^{\pi_{\theta}}(s_0, a_0)}_{(1)} + \underbrace{\sum_{a_0} \pi_{\theta}(a_0 | s_0) \nabla \left(Q^{\pi_{\theta}}(s_0, a_0) \right)}_{(2)} \end{aligned}$$

$$\begin{aligned} (1) &= \sum_{a_0} \pi_{\theta}(a_0 | s_0) \cdot \nabla_{\theta} \left(\log \pi_{\theta}(a_0 | s_0) \right) \cdot Q^{\pi_{\theta}}(s_0, a_0) \\ &= \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} \left[\nabla_{\theta} \left(\log \pi_{\theta}(a_0 | s_0) \right) \cdot Q^{\pi_{\theta}}(s_0, a_0) \right] \end{aligned}$$

(P2) REINFORCE Policy Gradient (Cont.)

$$\begin{aligned}(2) &= \sum_{a_0} \pi_{\theta}(a_0 | s_0) \nabla (Q^{\pi_{\theta}}(s_0, a_0)) \\&= \sum_{a_0} \pi_{\theta}(a_0 | s_0) \nabla_{\theta} \left(r(s_0, a_0) + \gamma \sum_{s_1} P(s_1 | s_0, a_0) \cdot V^{\pi_{\theta}}(s_1) \right) \\&= \sum_{a_0} \pi_{\theta}(a_0 | s_0) \left(\gamma \sum_{s_1} P(s_1 | s_0, a_0) \cdot \nabla_{\theta} V^{\pi_{\theta}}(s_1) \right) \\&= \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} [\gamma \nabla_{\theta} V^{\pi_{\theta}}(s_1)]\end{aligned}$$

By combining (1) and (2):

$$\nabla_{\theta} V^{\pi_{\theta}}(s_0) = \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} [\nabla_{\theta} (\log \pi_{\theta}(a_0 | s_0)) \cdot Q^{\pi_{\theta}}(s_0, a_0)] + \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} [\gamma \nabla_{\theta} V^{\pi_{\theta}}(s_1)]$$

This is a recursion!

(P2) REINFORCE Policy Gradient (Cont.)

Given the recursion:

$$\nabla_{\theta} V^{\pi_{\theta}}(s_0) = \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} \left[\nabla_{\theta} (\log \pi_{\theta}(a_0 | s_0)) \cdot Q^{\pi_{\theta}}(s_0, a_0) \right] + \mathbb{E}_{\tau \sim P_{s_0}^{\pi_{\theta}}} \left[\gamma \nabla_{\theta} V^{\pi_{\theta}}(s_1) \right]$$

- We can verify that:

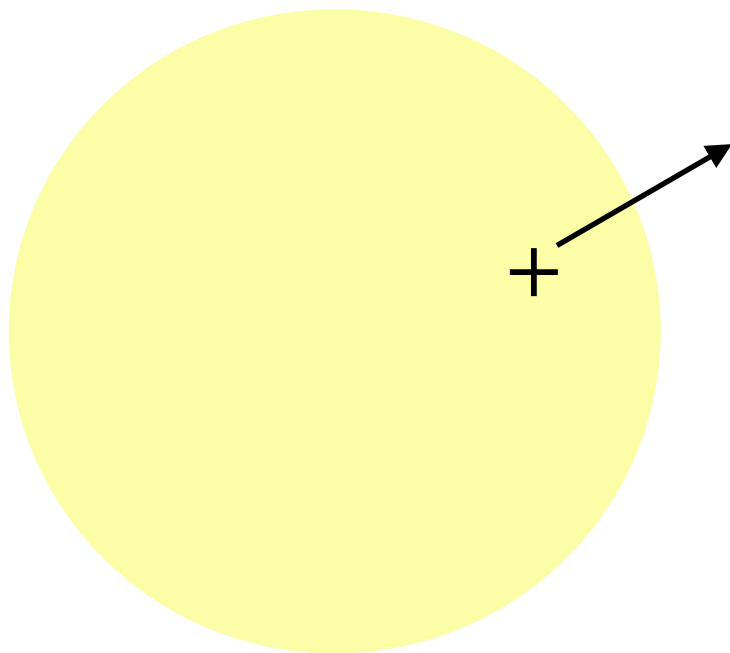
$$\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$$

(P3) Q-Value and Discounted State Visitation

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

- **Question:** How to interpret this expression?

Sample space of (s, a)



1. $Q(s, a) > 0$:

2. $Q(s, a) < 0$:

- **Question:** Why is this expression useful?

(P3) Q-Value and Discounted State Visitation (Cont.)

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

► **Proof idea:** Use the following lemma and (P2)

► **Lemma (From Trajectories to Visitation):**

For any function $f(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, we have

$$\mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t f(s_t, a_t) \right] = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta}}, a \sim \pi_{\theta}(\cdot | s)} [f(s, a)]$$

► **Proof:** Expand RHS and recollect terms to get LHS (HW1)

Computing $\nabla_{\theta} \log \pi_{\theta}(a | s)$: Linear Softmax Policy

- ▶ (P1)-(P3) all involve $\nabla_{\theta} \log \pi_{\theta}(a | s)$
- ▶ $\nabla_{\theta} \log \pi_{\theta}(a | s)$ is often called the **score function**
- ▶ **Example**: Linear softmax policy
 - ▶ Feature vector for each state-action pair $\phi(s, a)$
 - ▶ Probability of action is proportional to exponentiated weight

$$\pi_{\theta}(a | s) = \frac{e^{\phi(s,a)^T \theta}}{\sum_{a' \in \mathcal{A}} e^{\phi(s,a')^T \theta}}$$

- ▶ The score function is

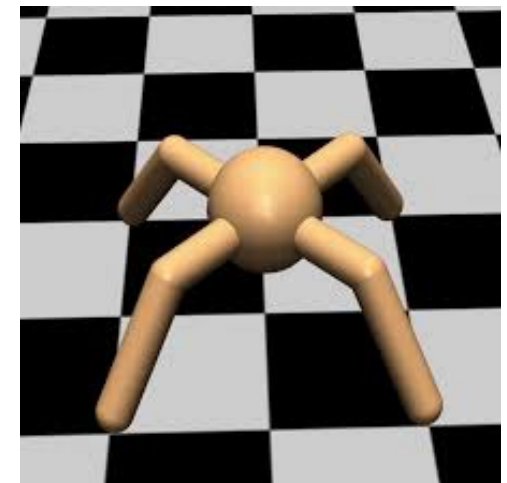
$$\nabla_{\theta} \log \pi_{\theta}(a | s) = \phi(s, a) - \mathbb{E}_{a \sim \pi_{\theta}(\cdot | s)}[\phi(s, a)]$$

Computing $\nabla_{\theta} \log \pi_{\theta}(a | s)$: Gaussian Policy

- ▶ **Example:** Gaussian policies for continuous actions

- ▶ $a \sim \mathcal{N}(\mu_{\theta}(s), \sigma^2)$
- ▶ Mean is a linear combination of state features $\mu_{\theta}(s) = \phi(s)^T \theta$
- ▶ Variance may be fixed σ^2 (or can also be parametrized)
- ▶ The score function is

$$\nabla_{\theta} \log \pi_{\theta}(a | s) = \frac{(a - \mu_{\theta}(s)) \phi(s)}{\sigma^2}$$



- ▶ **Question:** How about an NN-based Gaussian policy?

(P1) Total reward: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

(P2) REINFORCE: $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

(P3) Q-value and discounted state visitation:

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

Next Question: Is it easy to compute PG?

(P1)-(P3) all involve a highly complex expectation!

(This issue is quite common in many ML problems!)

Issues With (Exact) PG

- ▶ (Exact) PG Update (Use (P1) as an Example)

$$\begin{aligned}\theta_{k+1} &= \theta_k + \eta_k \cdot \nabla_{\theta} V^{\pi_{\theta}}(\mu)|_{\theta=\theta_k} \\ &= \theta_k + \eta_k \cdot \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]\end{aligned}$$

- ▶ **Question:** Any issues?

1. Distribution / statistics of τ is unknown (i.e., model-free setting)
2. Expectation usually involves multi-dimensional integral, which is computationally expensive

“Stochastic” Policy Gradient

- **Idea:** Use **sampling** to estimate expectation

$$\theta_{k+1} = \theta_k + \eta_k \cdot \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right] \quad (\text{Exact PG})$$



$$\theta_{k+1} = \theta_k + \eta_k \cdot \underbrace{G(\tau') \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a'_t | s'_t)}_{g(\theta_k, \tau')} \quad (\text{Stochastic PG})$$

where $\tau' \equiv (s'_0, a'_0, r'_1, s'_1, a'_1, r'_2, \dots)$

- $g(\theta_k, \tau')$ is an **unbiased** estimate of exact PG $\nabla_{\theta} V^{\pi_{\theta}}(\mu) |_{\theta=\theta_k}$
- An unbiased estimate can be constructed from 1 or multiple trajectories

More Generally: Stochastic Gradient Descent (SGD) for Stochastic Optimization

Stochastic Optimization:

$$\theta^* = \arg \max_{\theta \in \Theta} F(\theta), \text{ where } F(\theta) := \mathbb{E}_{\xi}[f(\theta; \xi)]$$

where ξ is the randomness (possibly unknown) in our problem

$$\begin{array}{ccc} \theta_{k+1} = \theta_k - \eta_k \cdot \mathbb{E}[\nabla_{\theta} f(\theta_k; \xi_k)] & \xrightarrow{\text{ }} & \theta_{k+1} = \theta_k - \eta_k \cdot g(\theta_k; \xi_k) \\ \text{(GD)} & & \text{(SGD)} \end{array}$$

- ▶ $g(\theta_k; \xi_k)$ can be constructed from 1 or multiple samples (mini-batch)
- ▶ **Advantage:** SGD has a low computational cost in each iteration

SGD: A Special Case of “Stochastic Approximation”

A STOCHASTIC APPROXIMATION METHOD¹

BY HERBERT ROBBINS AND SUTTON MONRO

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels x_1, x_2, \dots in such a way that x_n will tend to θ in probability.

2. Introduction. Let $M(x)$ be a given function and α a given constant such that the equation

$$(1) \quad M(x) = \alpha$$

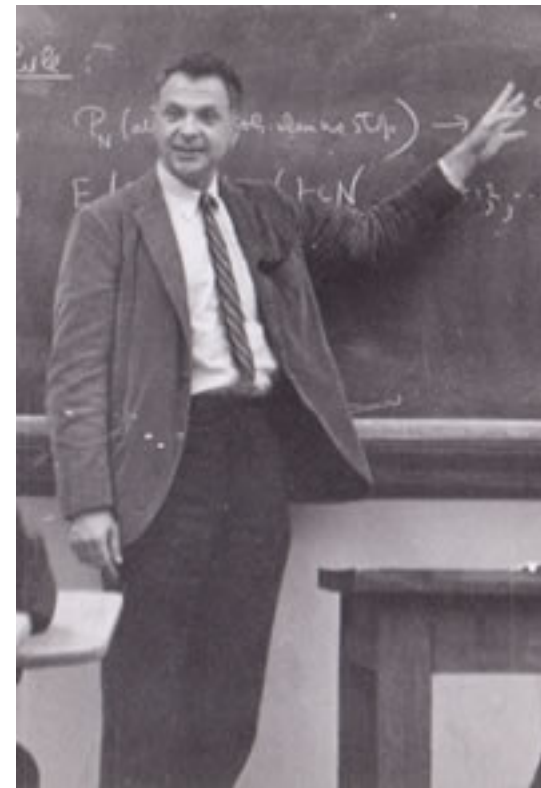
has a unique root $x = \theta$. There are many methods for determining the value of θ by successive approximation. With any such method we begin by choosing one or more values x_1, \dots, x_r more or less arbitrarily, and then successively obtain new values x_n as certain functions of the previously obtained x_1, \dots, x_{n-1} , the values $M(x_1), \dots, M(x_{n-1})$, and possibly those of the derivatives $M'(x_1), \dots, M'(x_{n-1})$, etc. If

$$(2) \quad \lim_{n \rightarrow \infty} x_n = \theta,$$

irrespective of the arbitrary initial values x_1, \dots, x_r , then the method is effective for the particular function $M(x)$ and value α . The speed of the convergence in (2) and the ease with which the x_n can be computed determine the practical utility of the method.

We consider a stochastic generalization of the above problem in which the nature of the function $M(x)$ is unknown to the experimenter. Instead, we suppose that to each value x corresponds a random variable $Y = Y(x)$ with distribution function $Pr[Y(x) \leq y] = H(y | x)$, such that

$$(3) \quad M(x) = \int_{-\infty}^{\infty} y dH(y | x)$$



Herbert Robbins



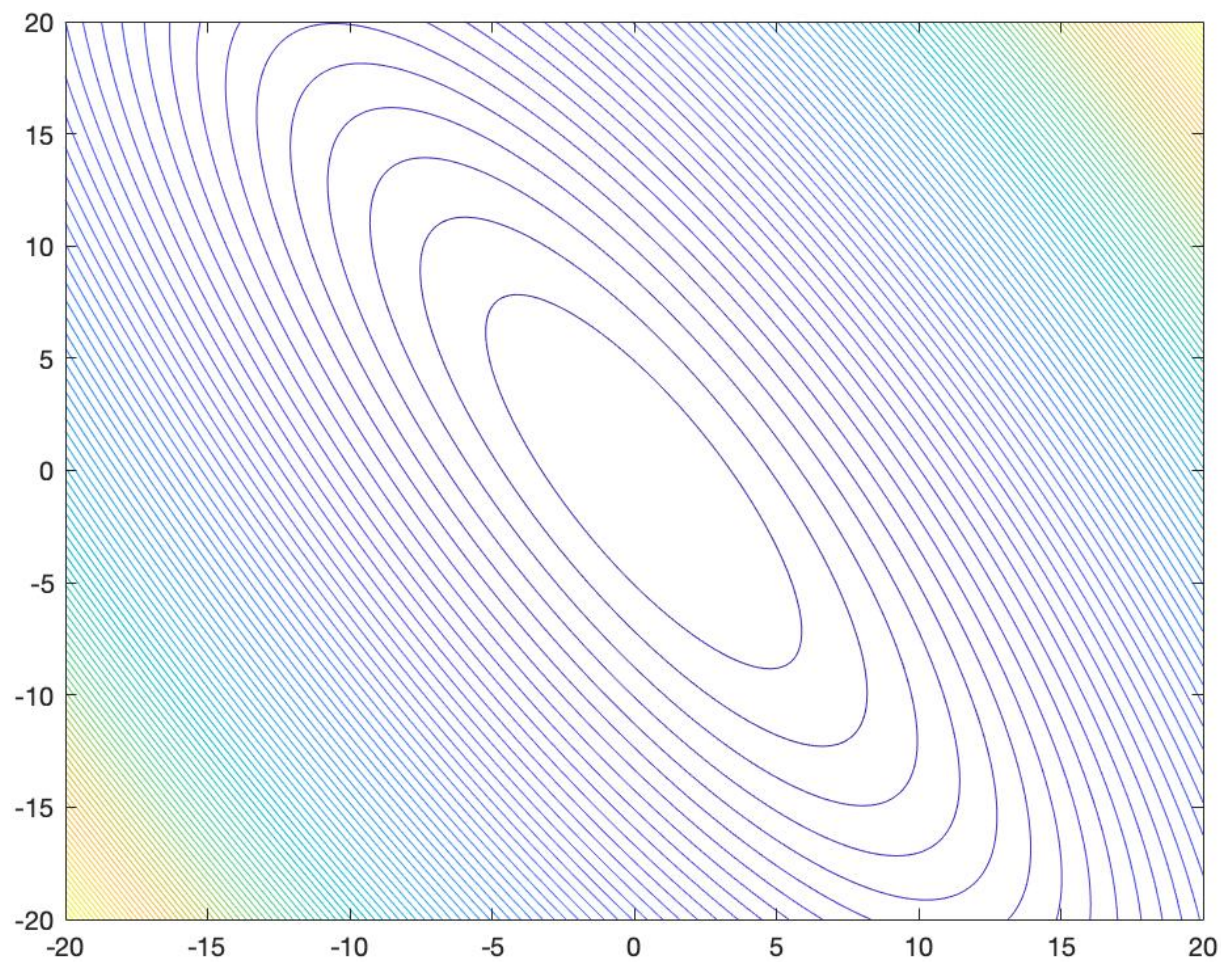
Sutton Monro

“Stochastic approximation” is the core of many RL methods, e.g.,
Q-learning and **TD learning**

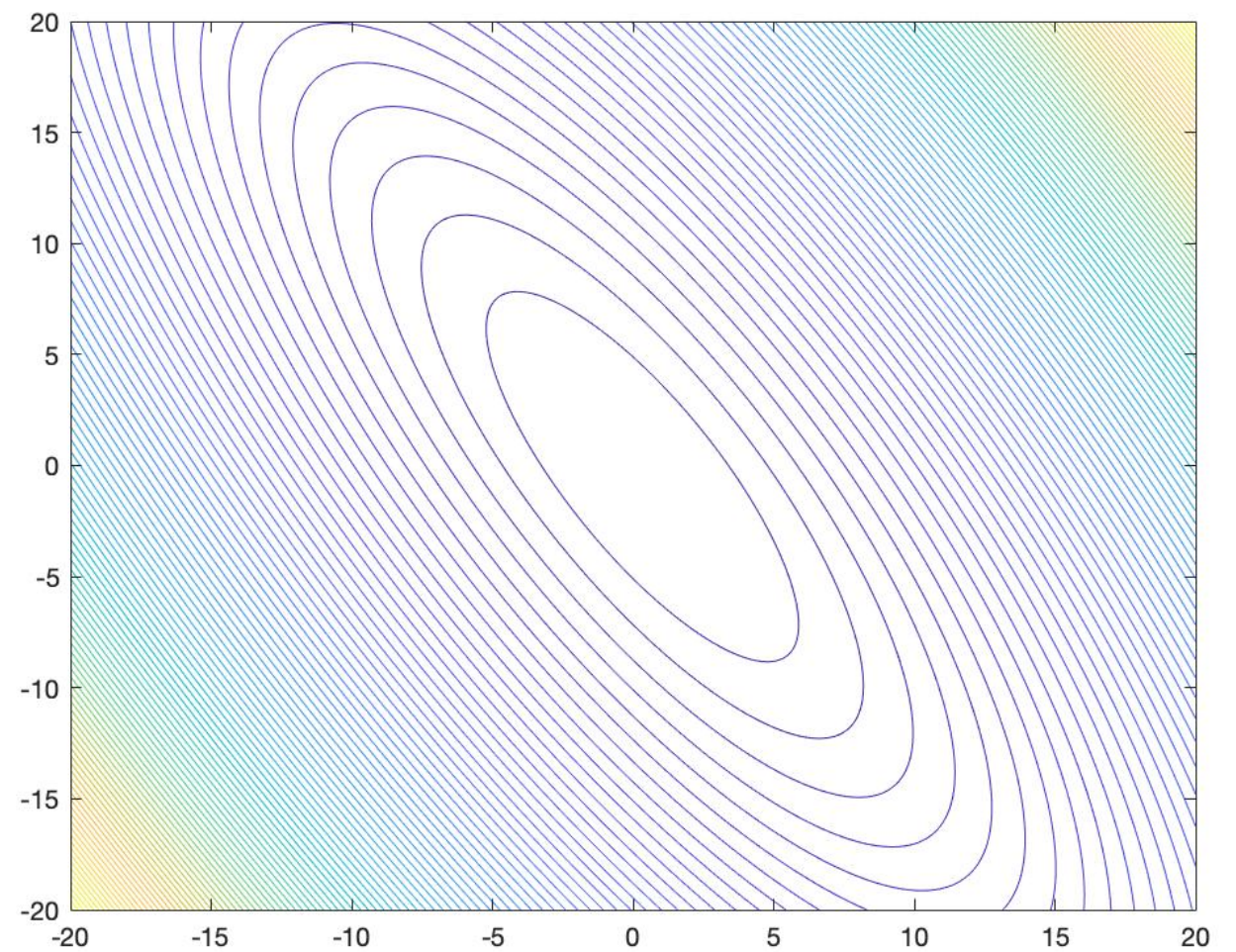
A seminal paper on stochastic approximation in 1951
(Cited for more than 11000 times)

Visualization: SGD vs GD

- ▶ SGD usually exhibits more “random” behavior than GD



GD



SGD

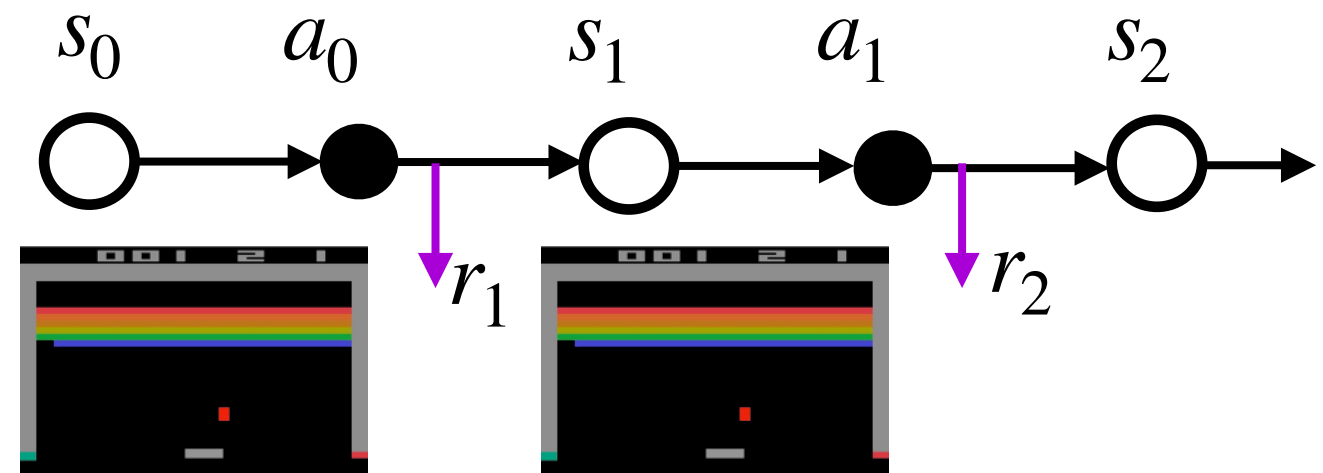
Let's combine SGD and PG!

Combining SGD and Policy Gradient

- ▶ Recall (P2): $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$
- ▶ In each iteration, we draw a trajectory $\tau = (s_0, a_0, r_1, s_1, a_1, \dots)$ under π_{θ} and μ , and then construct:

$$G_t(\tau) := \sum_{m=t}^{\infty} \gamma^m r_{m+1}$$

$$\hat{V}_{\tau} := \sum_{t=0}^{\infty} \gamma^t G_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$$



- ▶ **Question:** Is \hat{V}_{τ} an **unbiased** estimate of $\nabla_{\theta} V^{\pi_{\theta}}(\mu)$?

- ▶ (P2): $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t Q^{\pi_{\theta}}(s_t, a_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$
- ▶ Show that $\hat{\nabla}_{\tau}$ is an **unbiased** estimate of $\nabla_{\theta} V^{\pi_{\theta}}(\mu)$

$$G_t(\tau) := \sum_{m=t}^{\infty} \gamma^m r_{m+1}$$

$$\hat{\nabla}_{\tau} := \sum_{t=0}^{\infty} \gamma^t G_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$$

The REINFORCE Algorithm (Formally)

- **REINFORCE algorithm** (aka *Monte Carlo policy gradient*)

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

$$\begin{aligned}\theta_{k+1} &= \theta_k + \eta \cdot \hat{\nabla}_{\tau} \\ &= \theta_k + \eta \left(\sum_{t=0}^{\infty} \gamma^t G_t(\tau) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right)\end{aligned}$$

(Repeat Step 2 until termination)

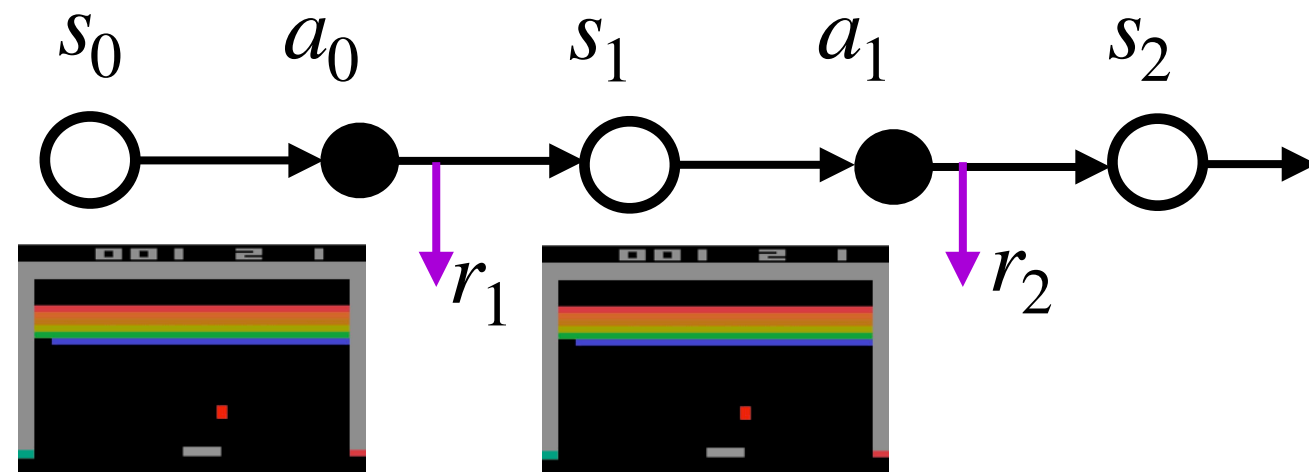
Can we design RL algorithms by using (P1) and (P3)?

An Alternative Monte-Carlo Policy Gradient Algorithm

► Recall (P1): $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \mathbb{E}_{\tau \sim P_{\mu}^{\pi_{\theta}}} \left[G(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right]$

Step 1: In each iteration k , draw a trajectory $\tau = (s_0, a_0, r_1, s_1, a_1, \dots)$ under π_{θ} and μ , and then construct:

$$G(\tau) := \sum_{t=0}^{\infty} \gamma^t r_t$$
$$\bar{\nabla}_{\tau} := R(\tau) \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$$



Step 2: Apply $\theta_{k+1} = \theta_k + \eta \cdot \bar{\nabla}_{\tau}$

-
- **Question:** Is $\bar{\nabla}_{\tau}$ an unbiased estimate of $\nabla_{\theta} V^{\pi_{\theta}}(\mu)$?
- **Question:** Any difference from REINFORCE?

How About Using (P3)?

► Recall (P3): $\nabla_{\theta} V^{\pi_{\theta}}(\mu) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta}}} \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[Q^{\pi_{\theta}}(s, a) \nabla_{\theta} \log \pi_{\theta}(a | s) \right]$

Step 1: In each iteration k , draw a batch B of n state-action pairs from $d_{\mu}^{\pi_{\theta}}$ and π_{θ} and construct

$$\tilde{\nabla}_{\tau} := \frac{1}{1-\gamma} \cdot \left(\frac{1}{n} \sum_{(s,a) \in B} Q^{\pi_{\theta}}(s, a) \nabla_{\theta} \log \pi_{\theta}(a | s) \right)$$

Step 2: Apply $\theta_{k+1} = \theta_k + \eta \cdot \tilde{\nabla}_{\tau}$

► **Question:** What does “draw samples from $d_{\mu}^{\pi_{\theta}}$ ” really mean?

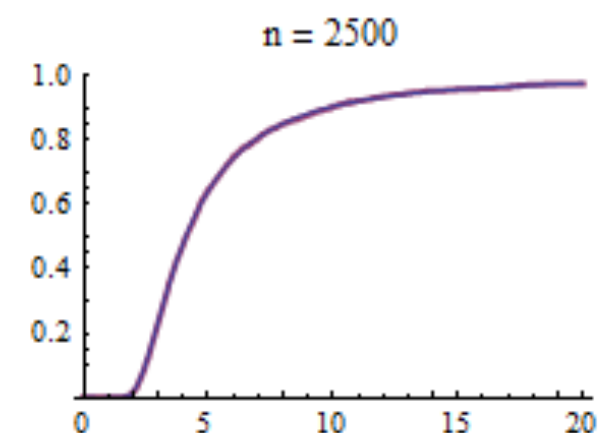
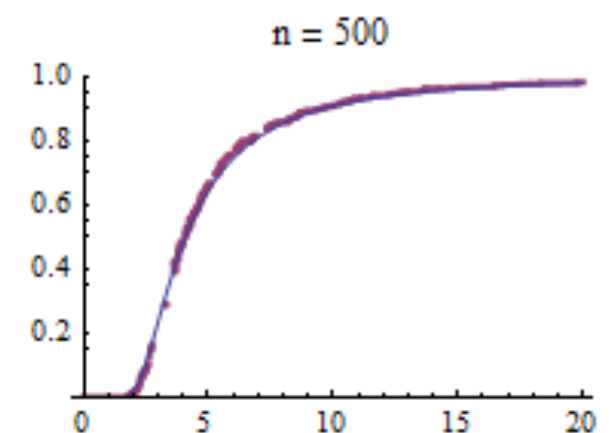
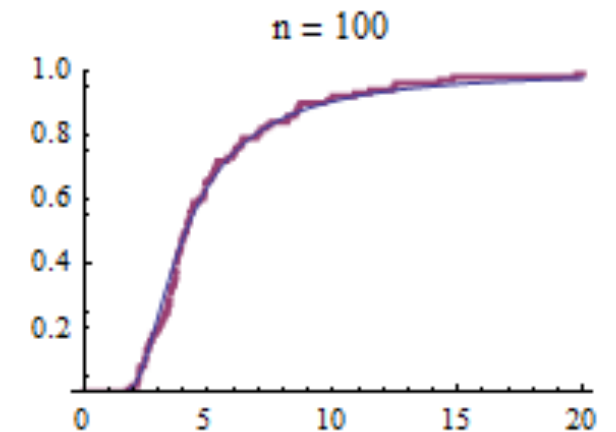
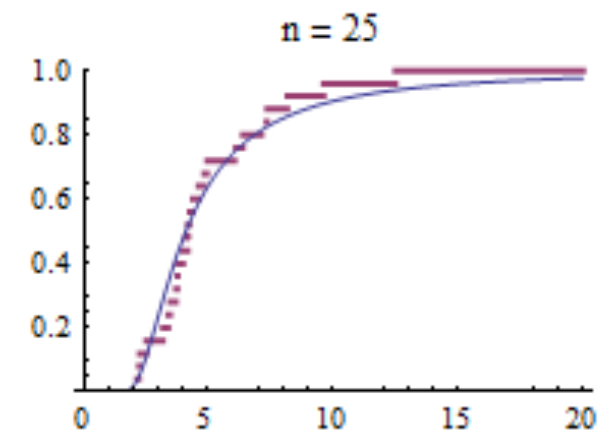
Are the samples i.i.d.?

An Untold Secret in RL Community...

- $\tilde{\nabla}_\tau$ for (P3) is actually NOT an unbiased estimator of the true PG!
- But for large n , $\tilde{\nabla}_\tau$ can still nicely approximate the true PG (Why?)

A Fundamental Property:

Empirical distribution uniformly approximates the true distribution!



Gilvenko-Cantelli Theorem (Formally)

Empirical distribution uniformly approximates the true distribution

- ▶ Let $\{X_n, n \geq 1\}$ be a sequence of i.i.d. random variables with a common CDF F

- ▶ Define empirical CDF as $\hat{F}_n(x) := \frac{1}{n} \sum_{i=1}^n I\{X_i \leq x\}$

- ▶ Define $D_n := \sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)|$

- ▶ **Gilvenko-Cantelli Theorem:**

$$D_n \rightarrow 0, \text{ as } n \rightarrow \infty$$

- ▶ This result could be extended to Markov chains

Yet Another Untold Secret in RL Community...

- RL people usually ignore the effect of γ on $d_{\mu}^{\pi_{\theta}}$ (which is not theoretically justified)

Is the Policy Gradient a Gradient?

Chris Nota

College of Information and Computer Sciences
University of Massachusetts Amherst
cnota@cs.umass.edu

ABSTRACT

The policy gradient theorem describes the gradient of the expected discounted return with respect to an agent’s policy parameters. However, most policy gradient methods drop the discount factor from the state distribution and therefore do not optimize the discounted objective. What do they optimize instead? This has been an open question for several years, and this lack of theoretical clarity has lead to an abundance of misstatements in the literature. We answer this question by proving that the update direction approximated by most methods is not the gradient of any function. Further, we argue that algorithms that follow this direction are not guaranteed to converge to a “reasonable” fixed point by constructing a counterexample wherein the fixed point is globally *pessimal* with respect to both the discounted and undiscounted objectives. We motivate this work by surveying the literature and showing that there remains a widespread misunderstanding regarding discounted policy gradient methods, with errors present even in highly-cited papers published at top conferences.

Philip S. Thomas

College of Information and Computer Sciences
University of Massachusetts Amherst
pthomas@cs.umass.edu

is pessimal, regardless of whether the discounted or undiscounted objective is considered.

The analysis in this paper applies to nearly all state-of-the-art policy gradient methods. In Section 6, we review all of the policy gradient algorithms included in the popular stable-baselines repository [9] and their associated papers, including A2C/A3C [13], ACER [28], ACKTR [30], DDPG [11], PPO [18], TD3 [6], TRPO [16], and SAC [8]. We motivate this choice in Section 6, but we note that all of these papers were published at top conferences¹ and have received hundreds or thousands of citations. We found that all of the implementations of the algorithms used the “incorrect” policy gradient that we discuss in this paper. While this is a valid algorithmic choice if properly acknowledged, we found that only *one* of the eight papers acknowledged this choice, while three of the papers made erroneous claims regarding the discounted policy gradient and others made claims that were misleading. The purpose of identifying these errors is not to criticize the authors or the algorithms, but to draw attention to the fact that confusion regarding the behavior of policy gradient algorithm exists at the very core of the RL community and has gone largely unnoticed by reviewers.



Philip Thomas