535514: Reinforcement Learning
Lecture 7 — Stochastic PG Algorithms

Ping-Chun Hsieh

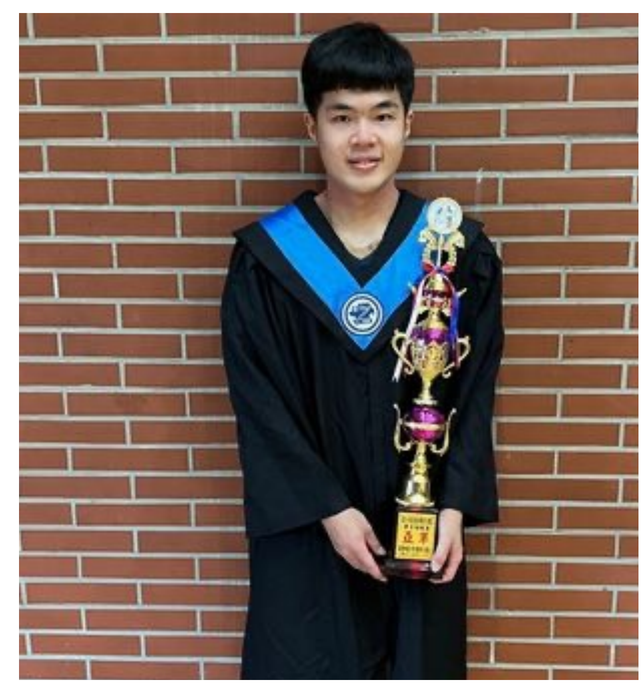
March 14, 2024

TA Lineup

林楷傑 (Kai-Jie Lin)
kjl0508.sc10@nycu.edu.tw

黃迺絜 (Nai-Chieh Huang)
naich.cs09@nycu.edu.tw

陳明宏 (Ming-Hong Chen)
andy7895as@gmail.com

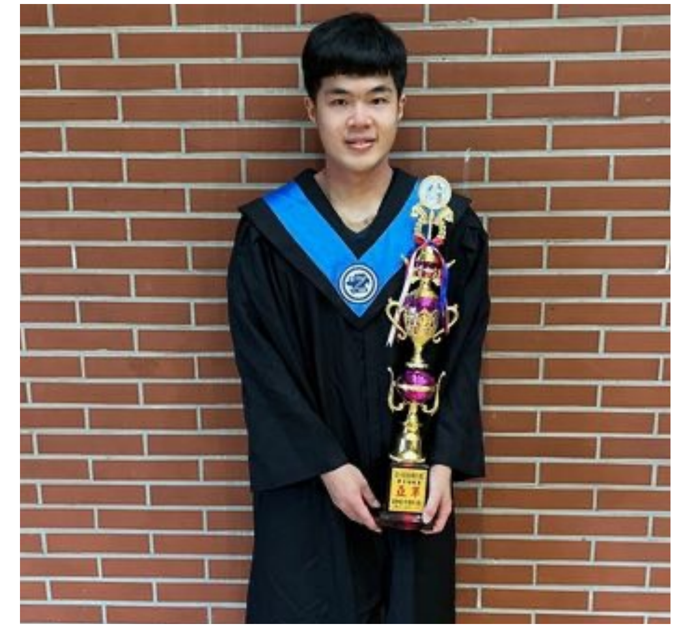


陳彥儒 (Yen-Ju Chen)
eric10400309@gmail.com

Mentor Lineup

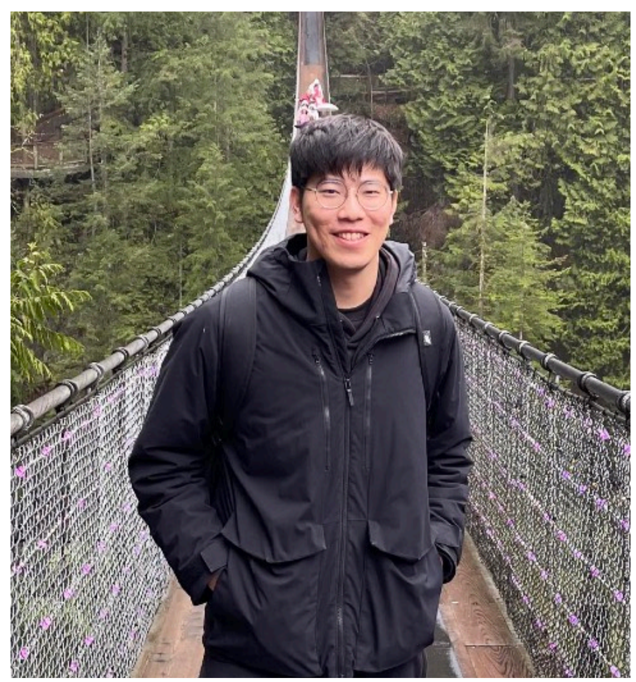
林楷傑 (Kai-Jie Lin)

黃迺絜 (Nai-Chieh Huang)



陳彥儒 (Yen-Ju Chen)

陳明宏 (Ming-Hong Chen)



王廣達 (Kuang-Da Wang)

陳盈圖 (Ying-Tu Chen)

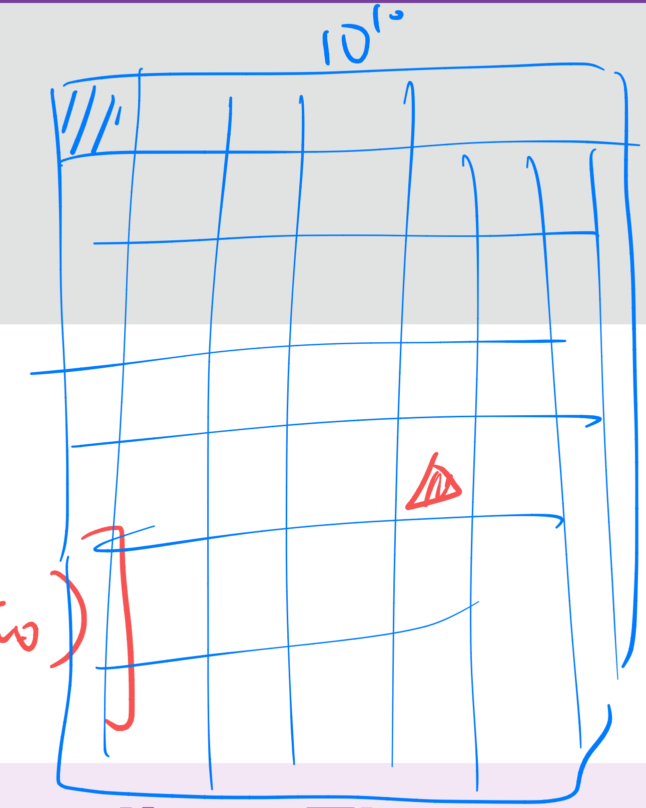
Team Implementation Project

- If you have successfully formed a team of 3-4 people, please fill out the form by tomorrow (3/15)
  - https://forms.gle/NARkSPvLL5VeUoyPA

- If you need help with team matching from the TAs, please fill out the form by tomorrow (3/15)
  - https://forms.gle/9f8MUGbEQ9xAA3k69

Quick Review: Policy Gradient

$\mu(s)>0$ for all s



- What is $V^{\pi_\theta}(\mu)$? $:= \mathbb{E}_{s_0\sim\mu}\left[V^{\pi_\theta}(s_0)\right]$
- What is $d_\mu^{\pi_\theta}$? $d_\mu^{\pi_\theta}(s) := (1-\gamma)\mathbb{E}_{s_0\sim\mu}\left[\sum \gamma^t \cdot P(s_t=s \mid s_0, \pi_\theta)\right]$

- **Expressions of Policy Gradient (aka Policy Gradient Theorem)**:

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

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

View this as $f(s,a)$

(P3) Q-value and discounted state visitation:

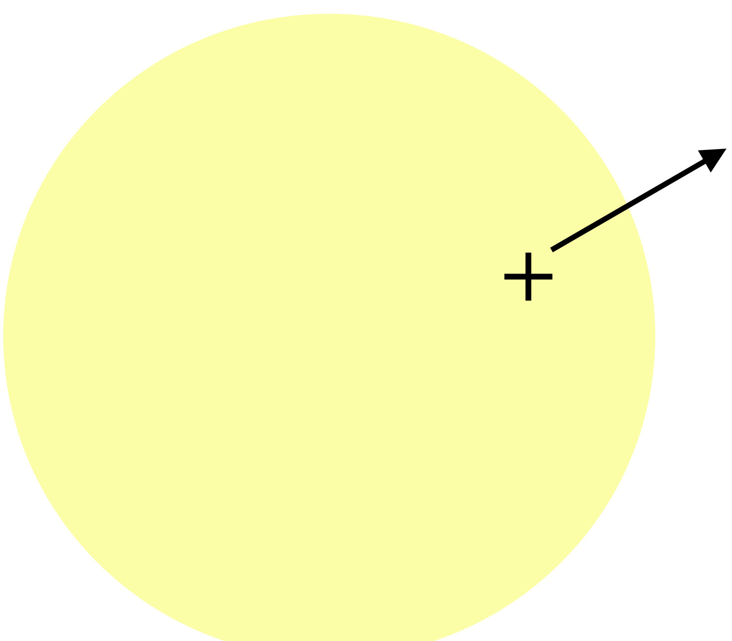
$$\nabla_\theta V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a \mid s)\right]$$

(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)



- $Q(s,a) > 0$:
- $Q(s,a) < 0$:

- 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 $f(s,a) : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$, we have

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

- 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"

$$\nabla_\theta \log \pi_\theta(a|s) = \nabla_\theta \left(\phi(s,a)^T\theta - \log \sum_{a'\in A} e^{\phi(s,a')^T\theta} \right)$$

$$= \phi(s,a) - \underbrace{\frac{1}{\sum_{a'\in A} e^{\phi(s,a')^T\theta}} \left(\sum_{a'} \phi(s,a')\cdot e^{\phi(s,a')^T\theta} \right)}$$

$$= \sum_{a'} \pi_\theta(a'|s)\cdot \phi(s,a')$$

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

- Example: Linear softmax policy
  - Feature vector for each state-action pair $\phi(s, a)$
  - Probability of action is proportional to exponentiated weight

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

  - The score function is

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

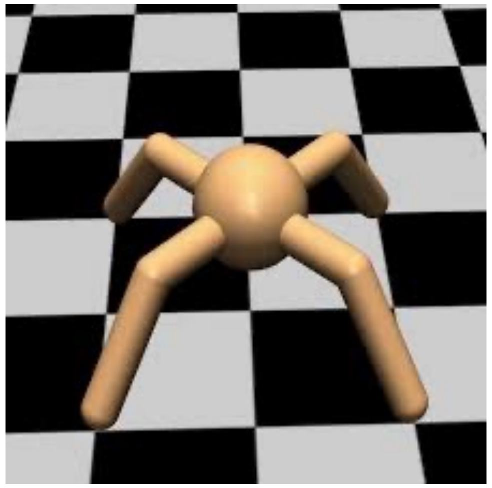
Computing $\nabla_\theta \log \pi_\theta(a \mid s)$: Gaussian Policy

(linear)

- **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

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



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

$\mu_\theta(s) \equiv NN_\theta(s)$

$s \to \boxed{NN\ (\theta)} \to \mu_\theta(s)$

$\pi_\theta(a|s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(a-\mu_\theta(s))^2}{\sigma^2}\right)$

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

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

(P3) Q-value and discounted state visitation:

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

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

step size

PG

- **(Exact) PG Update (Use (P1) as an Example)**:

$$\theta_{k+1} = \theta_k + \eta_k \cdot \nabla_\theta V^{\pi_\theta}(\mu)|_{\theta=\theta_k}$$

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

- Question: Any issues?

$P_\mu^{\pi_\theta}$ is unknown

1. Distribution / statistics of τ is unknown (i.e., model-free setting)

Even if $P_\mu^{\pi_\theta}$ is known,

2. Expectation usually involves multi-dimensional integral, which is computationally expensive

"Stochastic" Policy Gradient

- Idea: Use sampling to estimate expectation

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

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

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

- $g(\theta_k, \tau')$ 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

Unbiasedness

- **Definition (Unbiasedness)**:

  Let $\phi \in \mathbb{R}^d$ be some (unknown) real vector to be estimated, and let X be a random estimator of ϕ with distribution D.

  We say that X is an *unbiased* estimator of ϕ if

$$\mathbb{E}_{X \sim D}[X] = \phi$$

- Example: Let $X_1, \cdots, X_n$ be i.i.d. random variables from $\mathcal{N}(\mu, 1)$
  - Define $\bar{X} := (X_1 + \cdots + X_n)/n$. Then, is $\bar{X}$ an unbiased estimator of μ?

empirical mean

$$E[\bar{X}] = E\left[\frac{X_1 + X_2 + \cdots + X_n}{n}\right] = \frac{1}{n}\left(\underset{\mu}{E[X_1]} + \underset{\mu}{E[X_2]} + \cdots + \underset{\mu}{E[X_n]}\right)$$

$= \mu$ $\Rightarrow$ $\bar{X}$ is unbiased

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

Stochastic Optimization:

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

where ξ is the randomness in our problem

Stochastic Gradient Descent:

$$\theta_{k+1} = \theta_k - \eta_k \cdot \mathbb{E}[\nabla_\theta f(\theta_k; \xi)] \quad \Rightarrow \quad \theta_{k+1} = \theta_k - \eta_k \cdot g(\theta_k; \xi_k)$$

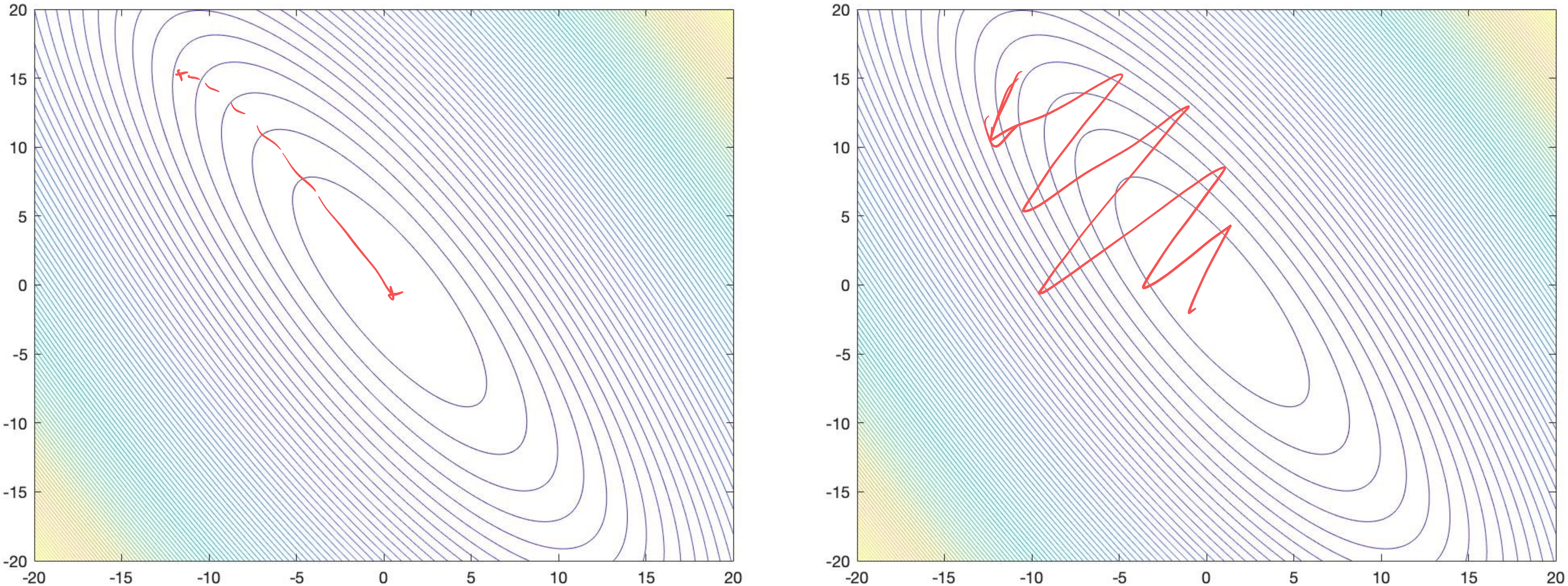
(GD) (SGD)

- $g(\theta_k; \xi_k)$ is an estimate of the true gradient (constructed from 1 or multiple samples)
- Advantage: SGD has a low computational cost in each iteration

H. Robbins and S. Monro, "A Stochastic Approximation Method," The Annals of Mathematical Statistics, 1951

Visualization: SGD vs GD

- SGD usually exhibits more “random” behavior than GD



GD

SGD

Almost All ML Problems are Stochastic Optimization Problems!

Policy Optimization in RL:

$$\max_{\theta} V^{\pi_\theta}(\mu)$$

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

Regression / Classification:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}\left[\ell\left(f_\theta(x), y\right)\right]$$

where ℓ is some loss function

Fine-Tuning of Language Models:

$$\max_{\theta} \mathbb{E}_{x\sim D}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x, y)\right] - \beta D_{KL}\left(\pi_\theta(\,\cdot\,|x) \| \pi_{ref}(\,\cdot\,|x)\right)\right]$$

SGD: A Special Case of "Stochastic Approximation"

A STOCHASTIC APPROXIMATION METHOD[1]

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 $x_1, x_2, \cdots$ 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

$$M(x) = \alpha \tag{1}$$

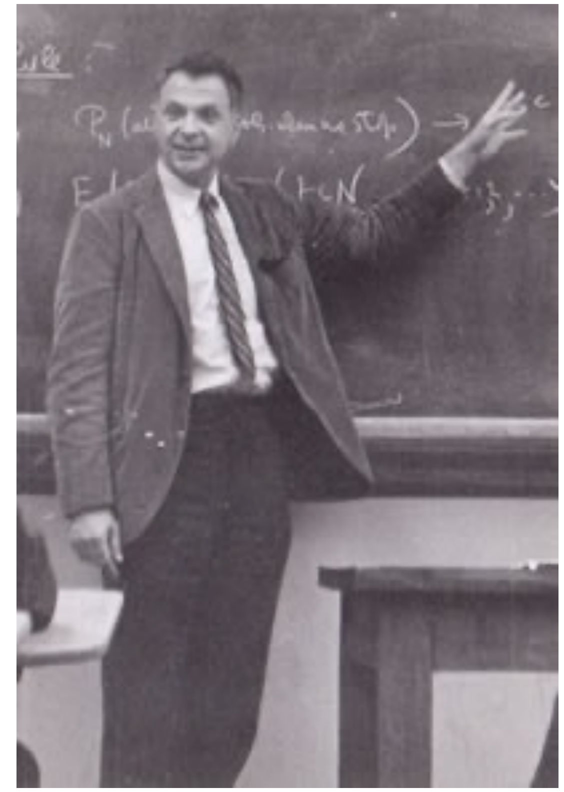
has a unique root $x = \theta$. There are many methods for determining the value of θ by successive approximation. With any such method we begin by choosing one or more values $x_1, \cdots, x_r$ more or less arbitrarily, and then successively obtain new values x_n as certain functions of the previously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

$$\lim_{n \to \infty} x_n = \theta, \tag{2}$$

irrespective of the arbitrary initial values $x_1, \cdots, x_r$, 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 $Pr[Y(x) \leq y] = H(y \mid x)$, such that

$$M(x) = \int_{-\infty}^{\infty} y \, dH(y \mid x) \tag{3}$$



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 12000 times)

Let's combine SGD and PG!

Combining SGD and Policy Gradient

$\mathbb{E}[G_t \mid s_t, a_t ; \pi_\theta]$

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

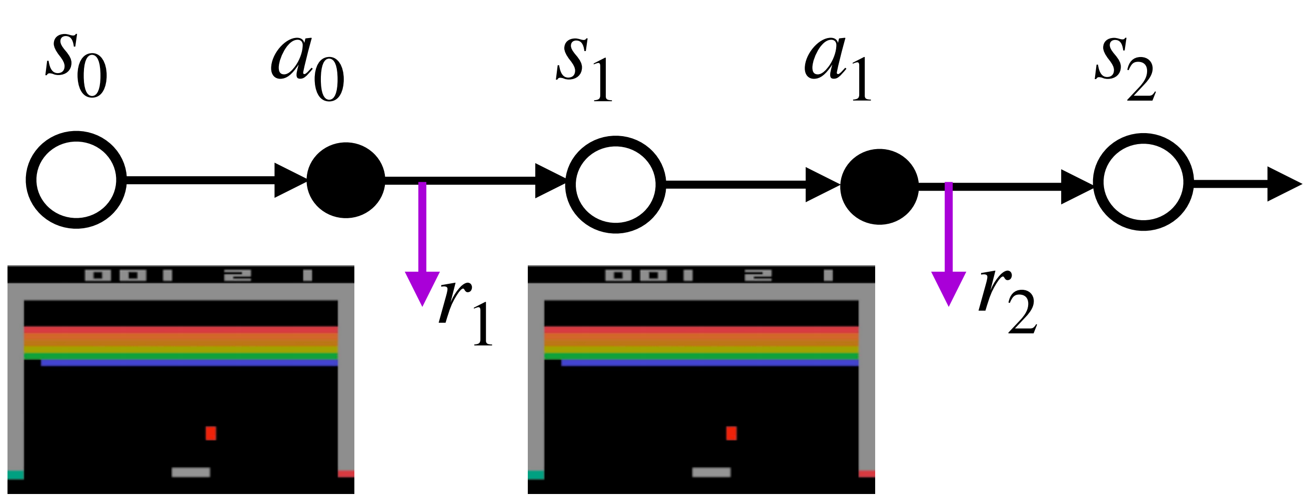
$a_0 \sim \pi_\theta(\cdot \mid s_0)$ $\quad a_1 \sim \pi_\theta(\cdot \mid s_1)$

- In each iteration, we draw a trajectory $\tau = (s_0, a_0, r_1, s_1, a_1 \cdots)$ under π_θ and μ, and then construct:

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

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

Stochastic PG



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

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

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

(Repeat Step 2 until termination)

Ronald J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," ML 1992

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

An Alternative Monte-Carlo Policy Gradient Algorithm

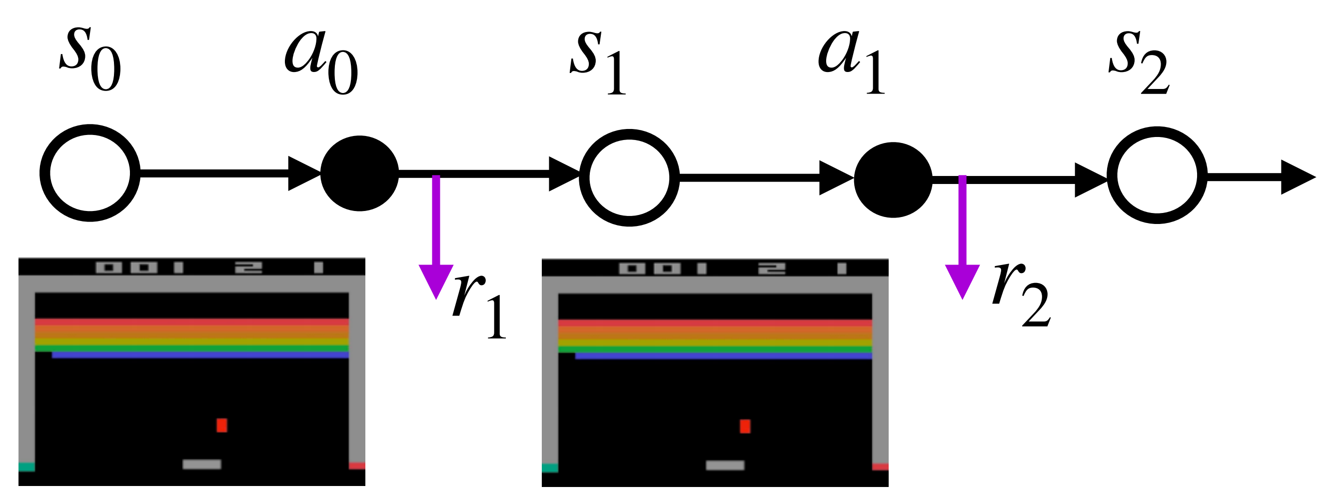
- Recall (P1): $\nabla_\theta V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}}\left[G(\tau) \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t | s_t)\right]$

Step 1: In each iteration k, draw a trajectory $\tau = (s_0, a_0, r_1, s_1, a_1 \cdots)$ under π_θ and μ, and then construct:

$$G(\tau) := \sum_{t=0}^{\infty} \gamma^t r_t$$

"Stochastic PG"

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



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

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

How About Using (P3)?

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

Step 1: In each iteration k, draw a batch B of n state-action pairs by following π_θ and construct

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

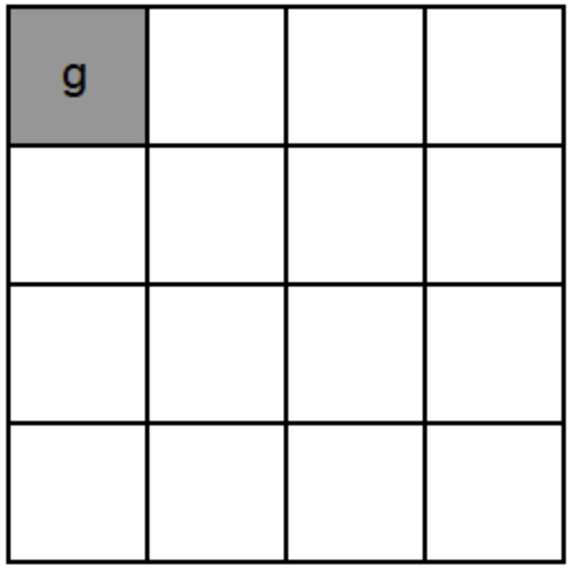
Step 2: Apply $\theta_{k+1} = \theta_k + \eta\cdot\tilde{\nabla}_\tau$

- Question: What does "draw samples by following π_θ" mean? Are the samples i.i.d.?

One 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!

(This is known as Gilvenko-Cantelli Theorem)

https://www.wolfram.com/mathematica/new-in-8/probability-and-statistics-solvers-and-properties/demonstrate-the-glivenko-cantelli-theorem.html

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 the 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:**

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

- This result could be directly extended to Markov chains

Stute and Schmann, "A General Gilvenko-Cantelli Theorem for Stationary Sequences of Random Observations," 1980

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

Philip S. Thomas
College of Information and Computer Sciences
University of Massachusetts Amherst
pthomas@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.

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 conferences[1] 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

Chris Nota and Philip Thomas, "Is the Policy Gradient a Gradient?," AAMAS 2020