# Policy Gradient Algorithms

Ashwin Rao

ICME, Stanford University

# Overview

# Why do we care about Policy Gradient (PG)?

- Let us review how we got here
- We started with Markov Decision Processes and Bellman Equations
- Next we studied several variants of DP and RL algorithms
- We noted that the idea of *Generalized Policy Iteration* (GPI) is key
- Policy Improvement step: $\pi(a|s)$ derived from $\text{argmax}_a Q(s, a)$
- How do we do argmax when action space is large or continuous?
- Idea: Do Policy Improvement step with a Gradient Ascent instead

# "Policy Improvement with a Gradient Ascent??"

- We want to find the Policy that fetches the "Best Expected Returns"
- Gradient Ascent on "Expected Returns" w.r.t params of Policy func
- So we need a func approx for (stochastic) Policy Func: $\pi(s, a; \theta)$
- In addition to the usual func approx for Action Value Func: $Q(s, a; w)$
- $\pi(s, a; \theta)$ func approx called *Actor*, $Q(s, a; w)$ func approx called *Critic*
- Critic parameters $w$ are optimized w.r.t $Q(s, a; w)$ loss function min
- Actor parameters $\theta$ are optimized w.r.t Expected Returns max
- We need to formally define "Expected Returns"
- But we already see that this idea is appealing for continuous actions
- GPI with Policy Improvement done as **Policy Gradient (Ascent)**

# Value Function-based and Policy-based RL

- Value Function-based
  - Learn Value Function (with a function approximation)
  - Policy is implicit - readily derived from Value Function (eg: $\epsilon$-greedy)
- Policy-based
  - Learn Policy (with a function approximation)
  - No need to learn a Value Function
- Actor-Critic
  - Learn Policy (Actor)
  - Learn Value Function (Critic)

# Advantages and Disadvantages of Policy Gradient approach

**Advantages:**

- Finds the best *Stochastic* Policy (Optimal Deterministic Policy, produced by other RL algorithms, can be unsuitable for POMDPs)
- Naturally *explores* due to Stochastic Policy representation
- Effective in high-dimensional or continuous action spaces
- Small changes in $\theta \Rightarrow$ small changes in $\pi$, and in state distribution
- This avoids the convergence issues seen in argmax-based algorithms

**Disadvantages:**

- Typically converge to a local optimum rather than a global optimum
- Policy Evaluation is typically inefficient and has high variance
- Policy Improvement happens in small steps $\Rightarrow$ slow convergence

# Notation

- Discount Factor $\gamma$
- Assume episodic with $0 \leq \gamma \leq 1$ or non-episodic with $0 \leq \gamma < 1$
- States $s_t \in \mathcal{S}$, Actions $a_t \in \mathcal{A}$, Rewards $r_t \in \mathbb{R}$, $\forall t \in \{0, 1, 2, \ldots\}$
- State Transition Probabilities $\mathcal{P}_{s,s'}^a = Pr(s_{t+1} = s' | s_t = s, a_t = a)$
- Expected Rewards $\mathcal{R}_s^a = E[r_t | s_t = s, a_t = a]$
- Initial State Probability Distribution $p_0 : \mathcal{S} \rightarrow [0, 1]$
- Policy Func Approx $\pi(s, a; \theta) = Pr(a_t = a | s_t = s, \theta), \theta \in \mathbb{R}^k$

PG coverage will be quite similar for non-discounted non-episodic, by considering average-reward objective (so we won't cover it)

# "Expected Returns" Objective

Now we formalize the "Expected Returns" Objective $J(\theta)$

$$J(\theta) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$$

Value Function $V^\pi(s)$ and Action Value function $Q^\pi(s,a)$ defined as:

$$V^\pi(s) = \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} r_k | s_t = s], \forall t \in \{0, 1, 2, \ldots\}$$

$$Q^\pi(s,a) = \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} r_k | s_t = s, a_t = a], \forall t \in \{0, 1, 2, \ldots\}$$

$$\text{Advantage Function } A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

Also, $p(s \rightarrow s', t, \pi)$ will be a key function for us - it denotes the probability of going from state $s$ to $s'$ in $t$ steps by following policy $\pi$

# Discounted State Visitation Measure

$$J(\theta) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t] = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\pi[r_t]$$

$$= \sum_{t=0}^{\infty} \gamma^t \int_{\mathcal{S}} (\int_{\mathcal{S}} p_0(s_0) \cdot p(s_0 \rightarrow s, t, \pi) \cdot ds_0) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot \mathcal{R}_s^a \cdot da \cdot ds$$

$$= \int_{\mathcal{S}} (\int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \cdot p_0(s_0) \cdot p(s_0 \rightarrow s, t, \pi) \cdot ds_0) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot \mathcal{R}_s^a \cdot da \cdot ds$$

## Definition

$$J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot \mathcal{R}_s^a \cdot da \cdot ds$$

where $\rho^\pi(s) = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \cdot p_0(s_0) \cdot p(s_0 \rightarrow s, t, \pi) \cdot ds_0$ is the key function (for PG) we'll refer to as *Discounted-Aggregate State-Visitation Measure*.

# Policy Gradient Theorem (PGT)

## Theorem

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s,a;\theta) \cdot Q^\pi(s,a) \cdot da \cdot ds$$

- Note: $\rho^\pi(s)$ depends on $\theta$, but there's no $\nabla_\theta \rho^\pi(s)$ term in $\nabla_\theta J(\theta)$
- So we can simply sample simulation paths, and at each time step, we calculate $(\nabla_\theta \log \pi(s,a;\theta)) \cdot Q^\pi(s,a)$ (probabilities implicit in paths)
- Note: $\nabla_\theta \log \pi(s,a;\theta)$ is Score function (Gradient of log-likelihood)
- We will estimate $Q^\pi(s,a)$ with a function approximation $Q(s,a;w)$
- We will later show how to avoid the estimate bias of $Q(s,a;w)$
- This numerical estimate of $\nabla_\theta J(\theta)$ enables **Policy Gradient Ascent**
- Let us look at the score function of some canonical $\pi(s,a;\theta)$

# Canonical $\pi(s, a; \theta)$ for finite action spaces

- For finite action spaces, we often use Softmax Policy
- $\theta$ is an $n$-vector $(\theta_1, \ldots, \theta_n)$
- Features vector $\phi(s, a) = (\phi_1(s, a), \ldots, \phi_n(s, a))$ for all $s \in \mathcal{S}, a \in \mathcal{A}$
- Weight actions using linear combinations of features: $\theta^T \cdot \phi(s, a)$
- Action probabilities proportional to exponentiated weights:

$$\pi(s, a; \theta) = \frac{e^{\theta^T \cdot \phi(s,a)}}{\sum_b e^{\theta^T \cdot \phi(s,b)}} \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}$$

- The score function is:

$$\nabla_\theta \log \pi(s, a; \theta) = \phi(s, a) - \sum_b \pi(s, b; \theta) \cdot \phi(s, b) = \phi(s, a) - \mathbb{E}_\pi[\phi(s, \cdot)]$$

# Canonical $\pi(s, a; \theta)$ for continuous action spaces

- For continuous action spaces, we often use Gaussian Policy
- $\theta$ is an $n$-vector $(\theta_1, \ldots, \theta_n)$
- State features vector $\phi(s) = (\phi_1(s), \ldots, \phi_n(s))$ for all $s \in \mathcal{S}$
- Gaussian Mean is a linear combination of state features $\theta^T \cdot \phi(s)$
- Variance may be fixed $\sigma^2$, or can also be parameterized
- Policy is Gaussian, $a \sim \mathcal{N}(\theta^T \cdot \phi(s), \sigma^2)$ for all $s \in \mathcal{S}$
- The score function is:

$$\nabla_\theta \log \pi(s, a; \theta) = \frac{(a - \theta^T \cdot \phi(s)) \cdot \phi(s)}{\sigma^2}$$

# Proof of Policy Gradient Theorem

We begin the proof by noting that:

$$J(\theta) = \int_{\mathcal{S}} p_0(s_0) \cdot V^\pi(s_0) \cdot ds_0 = \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0$$

Calculate $\nabla_\theta J(\theta)$ by parts $\pi(s_0, a_0; \theta)$ and $Q^\pi(s_0, a_0)$

$$\begin{aligned}\nabla_\theta J(\theta) = &\int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0 \\ &+ \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \pi(s_0, a_0; \theta) \cdot \nabla_\theta Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0\end{aligned}$$

# Proof of Policy Gradient Theorem

Now expand $Q^\pi(s_0, a_0)$ as $\mathcal{R}_{s_0}^{a_0} + \int_{\mathcal{S}} \gamma \cdot \mathcal{P}_{s_0,s_1}^{a_0} \cdot V^\pi(s_1) \cdot ds_1$ (Bellman)

$$= \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0$$
$$+ \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \pi(s_0, a_0; \theta) \cdot \nabla_\theta(\mathcal{R}_{s_0}^{a_0} + \int_{\mathcal{S}} \gamma \cdot \mathcal{P}_{s_0,s_1}^{a_0} \cdot V^\pi(s_1) \cdot ds_1) \cdot da_0 \cdot ds_0$$

Note: $\nabla_\theta \mathcal{R}_{s_0}^{a_0} = 0$, so remove that term

$$= \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a) \cdot da_0 \cdot ds_0$$
$$+ \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \pi(s_0, a_0; \theta) \cdot \nabla_\theta(\int_{\mathcal{S}} \gamma \cdot \mathcal{P}_{s_0,s_1}^{a_0} \cdot V^\pi(s_1) \cdot ds_1) \cdot da_0 \cdot ds_0$$

# Proof of Policy Gradient Theorem

Now bring the $\nabla_\theta$ inside the $\int_\mathcal{S}$ to apply only on $V^\pi(s_1)$

$$= \int_\mathcal{S} p_0(s_0) \int_\mathcal{A} \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a) \cdot da_0 \cdot ds_0$$
$$+ \int_\mathcal{S} p_0(s_0) \int_\mathcal{A} \pi(s_0, a_0; \theta) \int_\mathcal{S} \gamma \cdot \mathcal{P}_{s_0, s_1}^{a_0} \cdot \nabla_\theta V^\pi(s_1) \cdot ds_1 \cdot da_0 \cdot ds_0$$

Now bring the outside $\int_\mathcal{S}$ and $\int_\mathcal{A}$ inside the inner $\int_\mathcal{S}$

$$= \int_\mathcal{S} p_0(s_0) \int_\mathcal{A} \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0$$
$$+ \int_\mathcal{S} (\int_\mathcal{S} \gamma \cdot p_0(s_0) \int_\mathcal{A} \pi(s_0, a_0; \theta) \cdot \mathcal{P}_{s_0, s_1}^{a_0} \cdot da_0 \cdot ds_0) \cdot \nabla_\theta V^\pi(s_1) \cdot ds_1$$

# Policy Gradient Theorem

Note that $\int_{\mathcal{A}} \pi(s_0, a_0; \theta) \cdot \mathcal{P}_{s_0,s_1}^{a_0} \cdot da_0 = p(s_0 \rightarrow s_1, 1, \pi)$

$$= \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \cdot \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0$$
$$+ \int_{\mathcal{S}} (\int_{\mathcal{S}} \gamma \cdot p_0(s_0) \cdot p(s_0 \rightarrow s_1, 1, \pi) \cdot ds_0) \cdot \nabla_\theta V^\pi(s_1) \cdot ds_1$$

Now expand $V^\pi(s_1)$ to $\int_{\mathcal{A}} \pi(s_1, a_1; \theta) \cdot Q^\pi(s_1, a_1) \cdot da_1$

$$= \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \cdot \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da \cdot ds_0$$
$$+ \int_{\mathcal{S}} (\int_{\mathcal{S}} \gamma \cdot p_0(s_0) p(s_0 \rightarrow s_1, 1, \pi) ds_0) \cdot \nabla_\theta (\int_{\mathcal{A}} \pi(s_1, a_1; \theta) Q^\pi(s_1, a_1) da_1) ds_1$$

# Proof of Policy Gradient Theorem

We are now back to when we started calculating gradient of $\int_{\mathcal{A}} \pi \cdot Q^\pi \cdot da$. Follow the same process of splitting $\pi \cdot Q^\pi$, then Bellman-expanding $Q^\pi$ (to calculate its gradient), and iterate.

$$= \int_{\mathcal{S}} p_0(s_0) \int_{\mathcal{A}} \cdot \nabla_\theta \pi(s_0, a_0; \theta) \cdot Q^\pi(s_0, a_0) \cdot da_0 \cdot ds_0$$
$$+ \int_{\mathcal{S}} \int_{\mathcal{S}} \gamma p_0(s_0) p(s_0 \rightarrow s_1, 1, \pi) ds_0 ( \int_{\mathcal{A}} \nabla_\theta \pi(s_1, a_1; \theta) Q^\pi(s_1, a_1) da_1 + \ldots ) ds_1$$

This iterative process leads us to:

$$= \sum_{t=0}^{\infty} \int_{\mathcal{S}} \int_{\mathcal{S}} \gamma^t \cdot p_0(s_0) \cdot p(s_0 \rightarrow s_t, t, \pi) \cdot ds_0 \int_{\mathcal{A}} \nabla_\theta \pi(s_t, a_t; \theta) \cdot Q^\pi(s_t, a_t) \cdot da_t \cdot ds_t$$

# Proof of Policy Gradient Theorem

Bring $\sum_{t=0}^{\infty}$ inside the two $\int_{\mathcal{S}}$, and note that $\int_{\mathcal{A}} \nabla_\theta \pi(s_t, a_t; \theta) \cdot Q^\pi(s_t, a_t) \cdot da_t$ is independent of $t$.

$$= \int_{\mathcal{S}} \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \cdot p_0(s_0) \cdot p(s_0 \rightarrow s, t, \pi) \cdot ds_0 \int_{\mathcal{A}} \nabla_\theta \pi(s, a; \theta) \cdot Q^\pi(s, a) \cdot da \cdot ds$$

Reminder that $\int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t \cdot p_0(s_0) \cdot p(s_0 \rightarrow s, t, \pi) \cdot ds_0 \stackrel{\text{def}}{=} \rho^\pi(s)$. So,

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s, a; \theta) \cdot Q^\pi(s, a) \cdot da \cdot ds$$

$\mathbb{Q}.\mathbb{E}.\mathbb{D}.$

# Monte-Carlo Policy Gradient (REINFORCE Algorithm)

- Update $\theta$ by stochastic gradient ascent using PGT
- Using $G_t = \sum_{k=t}^{T} \gamma^{k-t} \cdot r_k$ as an unbiased sample of $Q^\pi(s_t, a_t)$

$$\Delta\theta_t = \alpha \cdot \gamma^t \cdot \nabla_\theta \log \pi(s_t, a_t; \theta) \cdot G_t$$

**Algorithm 4.1:** $\mathrm{REINFORCE}(\cdot)$

Initialize $\theta$ arbitrarily
**for** each episode $\{s_0, a_0, r_0, \ldots, s_T, a_T, r_T\} \sim \pi(\cdot, \cdot; \theta)$

$$\textbf{do} \begin{cases} \textbf{for } t \leftarrow 0 \textbf{ to } T \\ \quad \textbf{do} \begin{cases} G \leftarrow \sum_{k=t}^{T} \gamma^{k-t} \cdot r_k \\ \theta \leftarrow \theta + \alpha \cdot \gamma^t \cdot \nabla_\theta \log \pi(s_t, a_t; \theta) \cdot G \end{cases} \end{cases}$$

**return** $(\theta)$

# Reducing Variance using a Critic

- Monte Carlo Policy Gradient has high variance
- We use a Critic $Q(s, a; w)$ to estimate $Q^\pi(s, a)$
- Actor-Critic algorithms maintain two sets of parameters:
  - Critic updates parameters $w$ to approximate $Q$-function for policy $\pi$
  - Critic could use any of the algorithms we learnt earlier:
    - Monte Carlo policy evaluation
    - Temporal-Difference Learning
    - $TD(\lambda)$ based on Eligibility Traces
    - Could even use LSTD (if critic function approximation is linear)
  - Actor updates policy parameters $\theta$ in direction suggested by Critic
  - This is Approximate Policy Gradient due to *Bias* of Critic

$$\nabla_\theta J(\theta) \approx \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s, a; \theta) \cdot Q(s, a; w) \cdot da \cdot ds$$

# So what does the algorithm look like?

- Generate a sufficient set of simulation paths $s_0, a_0, r_0, s_1, a_1, r_1, \ldots$
- $s_0$ is sampled from the distribution $p_0(\cdot)$
- $a_t$ is sampled from $\pi(s_t, \cdot; \theta)$
- $s_{t+1}$ sampled from transition probs and $r_{t+1}$ from reward func
- At each time step $t$, update $w$ proportional to gradient of appropriate (MC or TD-based) loss function of $Q(s, a; w)$
- Sum $\gamma^t \cdot \nabla_\theta \log \pi(s_t, a_t; \theta) \cdot Q(s_t, a_t; w)$ over $t$ and over paths
- Update $\theta$ using this (biased) estimate of $\nabla_\theta J(\theta)$
- Iterate with a new set of simulation paths ...

# Reducing Variance with a Baseline

- We can reduce variance by subtracting a baseline function $B(s)$ from $Q(s, a; w)$ in the Policy Gradient estimate
- This means at each time step, we replace $\gamma^t \cdot \nabla_\theta \log \pi(s_t, a_t; \theta) \cdot Q(s_t, a_t; w)$ with $\gamma^t \cdot \nabla_\theta \log \pi(s_t, a_t; \theta) \cdot (Q(s_t, a_t; w) - B(s))$
- Note that Baseline function $B(s)$ is only a function of $s$ (and not $a$)
- This ensures that subtracting Baseline $B(s)$ does not add bias

$$\int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s, a; \theta) \cdot B(s) \cdot da \cdot ds$$
$$= \int_{\mathcal{S}} \rho^\pi(s) \cdot B(s) \cdot \nabla_\theta (\int_{\mathcal{A}} \pi(s, a; \theta) \cdot da) \cdot ds = 0$$

# Using State Value function as Baseline

- A good baseline $B(s)$ is state value function $V(s;v)$
- Rewrite Policy Gradient algorithm using advantage function estimate

$$A(s,a;w,v) = Q(s,a;w) - V(s;v)$$

- Now the estimate of $\nabla_\theta J(\theta)$ is given by:

$$\int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s,a;\theta) \cdot A(s,a;w,v) \cdot da \cdot ds$$

- At each time step, we update both sets of parameters $w$ and $v$

# TD Error as estimate of Advantage Function

- Consider TD error $\delta^\pi$ for the *true* Value Function $V^\pi(s)$

$$\delta^\pi = r + \gamma V^\pi(s') - V^\pi(s)$$

- $\delta^\pi$ is an unbiased estimate of Advantage function $A^\pi(s,a)$

$$\mathbb{E}_\pi[\delta^\pi|s,a] = \mathbb{E}_\pi[r+\gamma V^\pi(s')|s,a] - V^\pi(s) = Q^\pi(s,a) - V^\pi(s) = A^\pi(s,a)$$

- So we can write Policy Gradient in terms of $\mathbb{E}_\pi[\delta^\pi|s,a]$

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s,a;\theta) \cdot \mathbb{E}_\pi[\delta^\pi|s,a] \cdot da \cdot ds$$

- In practice, we can use func approx for TD error (and sample):

$$\delta(s,r,s';v) = r + \gamma V(s';v) - V(s;v)$$

- This approach requires only one set of critic parameters $v$

# TD Error can be used by both Actor and Critic

**Algorithm 4.2:** ACTOR-CRITIC-TD-ERROR($\cdot$)

Initialize Policy params $\theta \in \mathbb{R}^m$ and State VF params $v \in \mathbb{R}^n$ arbitrarily
**for** each episode

$$
\textbf{do} \begin{cases} \text{Initialize } s \text{ (first state of episode)} \\ P \leftarrow 1 \\ \textbf{while } s \text{ is not terminal} \\ \quad \textbf{do} \begin{cases} a \sim \pi(s, \cdot; \theta) \\ \text{Take action } a, \text{ observe } r, s' \\ \delta \leftarrow r + \gamma V(s'; v) - V(s; v) \\ v \leftarrow v + \alpha_v \cdot \delta \cdot \nabla_v V(s; v) \\ \theta \leftarrow \theta + \alpha_\theta \cdot P \cdot \delta \cdot \nabla_\theta \log \pi(s, a; \theta) \\ P \leftarrow \gamma P \\ s \leftarrow s' \end{cases} \end{cases}
$$

# Using Eligibility Traces for both Actor and Critic

**Algorithm 4.3:** ACTOR-CRITIC-ELIGIBILITY-TRACES($\cdot$)

Initialize Policy params $\theta \in \mathbb{R}^m$ and State VF params $v \in \mathbb{R}^n$ arbitrarily
**for** each episode
**do**
- Initialize $s$ (first state of episode)
- $z_\theta, z_v \leftarrow 0$ ($m$ and $n$ components eligibility trace vectors)
- $P \leftarrow 1$
- **while** $s$ is not terminal
  **do**
  - $a \sim \pi(s, \cdot; \theta)$
  - Take action $a$, observe $r, s'$
  - $\delta \leftarrow r + \gamma V(s'; v) - V(s; v)$
  - $z_v \leftarrow \gamma \cdot \lambda_v \cdot z_v + \nabla_v V(s; v)$
  - $z_\theta \leftarrow \gamma \cdot \lambda_\theta \cdot z_\theta + P \cdot \nabla_\theta \log \pi(s, a; \theta)$
  - $v \leftarrow v + \alpha_v \cdot \delta \cdot z_v$
  - $\theta \leftarrow \theta + \alpha_\theta \cdot \delta \cdot z_\theta$
  - $P \leftarrow \gamma P, s \leftarrow s'$

# Overcoming Bias

- We've learnt a few ways of how to reduce variance
- But we haven't discussed how to overcome bias
- All of the following substitutes for $Q^\pi(s,a)$ in PG have bias:
  - $Q(s,a;w)$
  - $A(s,a;w,v)$
  - $\delta(s,s',r;v)$
- Turns out there is indeed a way to overcome bias
- It is called the *Compatible Function Approximation Theorem*

# Compatible Function Approximation Theorem

## Theorem

*If the following two conditions are satisfied:*

1. *Critic gradient is* compatible *with the Actor score function*

$$\nabla_w Q(s, a; w) = \nabla_\theta \log \pi(s, a; \theta)$$

2. *Critic parameters $w$ minimize the following mean-squared error:*

$$\epsilon = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta)(Q^\pi(s, a) - Q(s, a; w))^2 \cdot da \cdot ds$$

*Then the Policy Gradient using critic $Q(s, a; w)$ is exact:*

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s, a; \theta) \cdot Q(s, a; w) \cdot da \cdot ds$$

# Proof of Compatible Function Approximation Theorem

For $w$ that minimizes

$$\epsilon = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot (Q^\pi(s, a) - Q(s, a; w))^2 \cdot da \cdot ds,$$

$$\int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot (Q^\pi(s, a) - Q(s, a; w)) \cdot \nabla_w Q(s, a; w) \cdot da \cdot ds = 0$$

But since $\nabla_w Q(s, a; w) = \nabla_\theta \log \pi(s, a; \theta)$, we have:

$$\int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot (Q^\pi(s, a) - Q(s, a; w)) \cdot \nabla_\theta \log \pi(s, a; \theta) \cdot da \cdot ds = 0$$

$$\begin{aligned} \text{Therefore, } & \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot Q^\pi(s, a) \cdot \nabla_\theta \log \pi(s, a; \theta) \cdot da \cdot ds \\ &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot Q(s, a; w) \cdot \nabla_\theta \log \pi(s, a; \theta) \cdot da \cdot ds \end{aligned}$$

# Proof of Compatible Function Approximation Theorem

$$\text{But } \nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s,a;\theta) \cdot Q^\pi(s,a) \cdot \nabla_\theta \log \pi(s,a;\theta) \cdot da \cdot ds$$

$$\begin{aligned}\text{So, } \nabla_\theta J(\theta) &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s,a;\theta) \cdot Q(s,a;w) \cdot \nabla_\theta \log \pi(s,a;\theta) \cdot da \cdot ds \\ &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \nabla_\theta \pi(s,a;\theta) \cdot Q(s,a;w) \cdot da \cdot ds\end{aligned}$$

$\mathbb{Q}.\mathbb{E}.\mathbb{D}.$

**This means with conditions (1) and (2) of Compatible Function Approximation Theorem, we can use the critic func approx $Q(s,a;w)$ and still have the exact Policy Gradient.**

# How to enable Compatible Function Approximation

A simple way to enable Compatible Function Approximation $\frac{\partial Q(s,a;w)}{\partial w_i} = \frac{\partial \log \pi(s,a;\theta)}{\partial \theta_i}, \forall i$ is to set $Q(s,a;w)$ to be linear in its features.

$$Q(s,a;w) = \sum_{i=1}^{n} \phi_i(s,a) \cdot w_i = \sum_{i=1}^{n} \frac{\partial \log \pi(s,a;\theta)}{\partial \theta_i} \cdot w_i$$

We note below that a compatible $Q(s,a;w)$ serves as an approximation of the advantage function.

$$\int_{\mathcal{A}} \pi(s,a;\theta) \cdot Q(s,a;w) \cdot da = \int_{\mathcal{A}} \pi(s,a;\theta) \cdot (\sum_{i=1}^{n} \frac{\partial \log \pi(s,a;\theta)}{\partial \theta_i} \cdot w_i) \cdot da$$

$$= \int_{\mathcal{A}} \cdot (\sum_{i=1}^{n} \frac{\partial \pi(s,a;\theta)}{\partial \theta_i} \cdot w_i) \cdot da = \sum_{i=1}^{n} (\int_{\mathcal{A}} \frac{\partial \pi(s,a;\theta)}{\partial \theta_i} \cdot da) \cdot w_i$$

$$= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} (\int_{\mathcal{A}} \pi(s,a;\theta) \cdot da) \cdot w_i = \sum_{i=1}^{n} \frac{\partial 1}{\partial \theta_i} \cdot w_i = 0$$

# Fisher Information Matrix

Denoting $[\frac{\partial \log \pi(s,a;\theta)}{\partial \theta_i}], i = 1, \ldots, n$ as the score column vector $SC(s, a; \theta)$ and assuming compatible linear-approx critic:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} \pi(s, a; \theta) \cdot (SC(s, a; \theta) \cdot SC(s, a; \theta)^T \cdot w) \cdot da \cdot ds \\
&= E_{s \sim \rho^\pi, a \sim \pi}[SC(s, a; \theta) \cdot SC(s, a; \theta)^T] \cdot w \\
&= FIM_{\rho^\pi, \pi}(\theta) \cdot w
\end{aligned}$$

where $FIM_{\rho_\pi, \pi}(\theta)$ is the Fisher Information Matrix w.r.t. $s \sim \rho^\pi, a \sim \pi$.

# Natural Policy Gradient

- Recall the idea of Natural Gradient from Numerical Optimization
- Natural gradient $\nabla_\theta^{nat} J(\theta)$ is the direction of optimal $\theta$ movement
- In terms of the KL-divergence metric (versus plain Euclidean norm)
- Natural gradient yields better convergence (we won't cover proof)

$$\text{Formally defined as: } \nabla_\theta J(\theta) = FIM_{\rho_\pi,\pi}(\theta) \cdot \nabla_\theta^{nat} J(\theta)$$

$$\text{Therefore, } \nabla_\theta^{nat} J(\theta) = w$$

**This compact result is great for our algorithm:**

- Update Critic params $w$ with the critic loss gradient (at step $t$) as:

$$\gamma^t \cdot (r_t + \gamma \cdot SC(s_{t+1}, a_{t+1}, \theta) \cdot w - SC(s_t, a_t, \theta) \cdot w) \cdot SC(s_t, a_t, \theta)$$

- Update Actor params $\theta$ in the direction equal to value of $w$