Policy Gradient Methods

- What are policy gradient reinforcement learning methods
- Properties of parameterized policies
- Policy gradient theorem
- Reinforce algorithm
- Stochastic policies
- Reducing variance with a critic
- Bias in actor-critic methods
- Learning the critic function
- Advantage Actor Critic (A2C) methods
- Summary

What Are Policy Gradient Methods?

Value Iteration

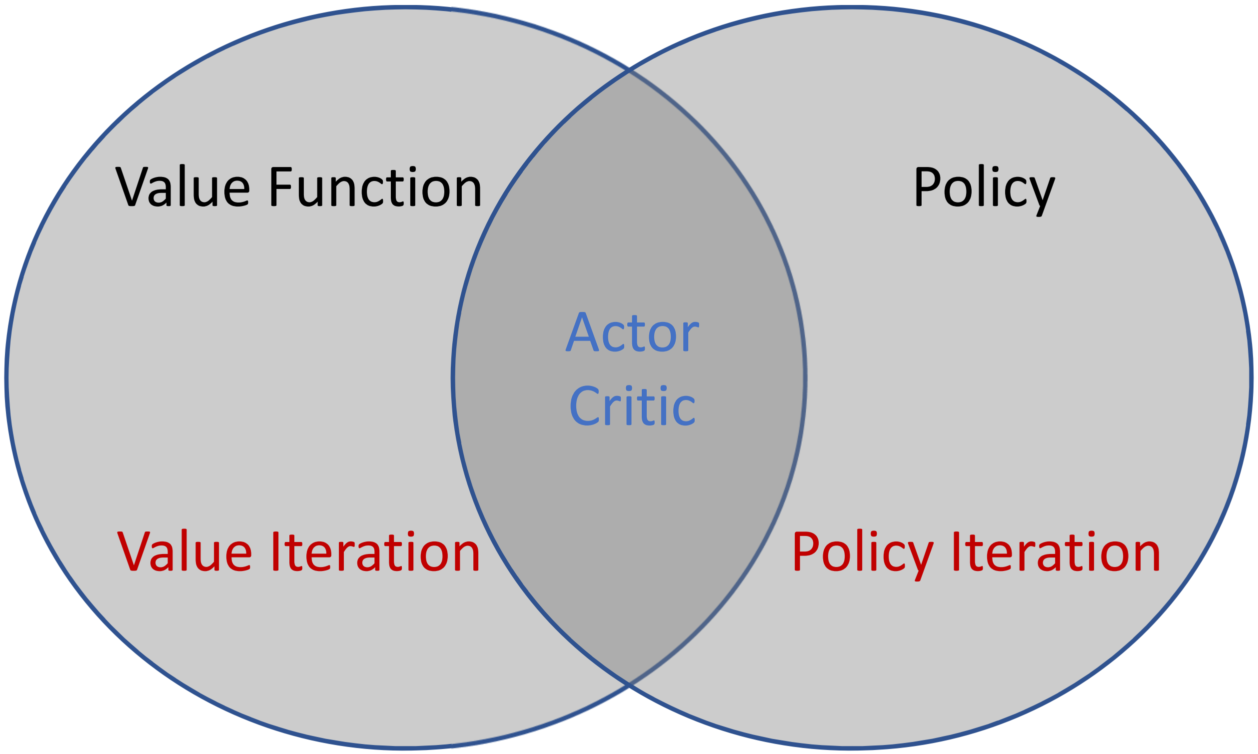
- Learned value function
- Implicit policy; ε-greedy

Policy Iteration

- No value function
- Learn policy

Actor-Critic

- Learn value function
- Learn policy



What Are Policy Gradient Methods?

- Previously, we parameterized value or action value function with parameters, w:

$$V(s_t) \approx V(s_t, \mathbf{w}_t)$$

$$Q(s_t, a_t) \approx Q(s_t, a_t, \mathbf{w}_t)$$

- e.g. action-value **function approximation**, with weights, **w**:

$$Q(S_t, A_t) = \hat{Q}(S_t, A_t, w_t)$$

- Learn weights with gradient

$$\hat{\nabla}_w q(S_t, A_t, \mathbf{w}_t)$$

What Are Policy Gradient Methods?

- Can **directly parameterize policy,** with parameters, θ:

$$\pi(a \mid s, \theta) = Pr\{A_t = a \mid S_t = s, \theta_\mathbf{t} = \theta\}$$

- Learn parameters with **policy gradient**:

$$\nabla_\theta \pi(a|S_t, \theta)$$

What Are Policy Gradient Methods?

Advantages of policy gradient methods

- **Improved convergence properties** in some cases, with greater sample efficiency
- Scalable
  - **High dimensional action spaces**
  - **Continuous action spaces**
- Supports stochastic policy
  - Previous methods used **greedy or ε-greedy** methods giving **deterministic policy**
  - But optimal policy can be **non-deteriministic, stochastic**

What Are Policy Gradient Methods?

Disadvantages of policy gradient methods

- Often converge to a **locally, rather than globally, optimal solution**
- Policy evaluation has **high variance** leading to inaccurate solutions

Properties of Parameterized Policies

Parameterized policies, $\pi(a \mid s, \theta)$, must:

- Be **differentiable everywhere**

  $\nabla_\theta \pi(a \mid s, \theta)$ must be **continuous and bounded** for $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

- **Never be deterministic**

  e.g. $0 > \pi(a \mid s, \theta) > 1.0, \ \forall\, a, \ \forall\, s$

  not $\pi(a \mid s, \theta) \in \{0, 1\}$, a binary action choice

Policy Gradient Theorem

Find an **analytic** representation of the **policy gradient**

- For episodic MDP, the Monte Carlo performance measure is the **loss function**:

$$J(\theta) = v_{\pi_\theta}(s_0)$$

Where s_0 is the initial state

- Given this loss function the policy gradient is then:

$$\nabla_\theta J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, q) \nabla_\theta \pi(a|S_t, \theta)$$

$$= \mathbb{E}_{\pi_\theta}\Big[\sum_a q_\pi(s, q) \nabla_\theta \pi(a|S_t, \theta) \Big]$$

Policy Gradient Theorem

How to compute $\nabla_\theta \pi(a|S_t, \theta)$?

- Start with the likelihood ratio:

$$\nabla_\theta \pi(a|S_t, \theta) = \pi(a|S_t, \theta) \frac{\nabla_\theta \pi(a|S_t, \theta)}{\pi(a|S_t, \theta)}$$

- Use the identity to define the **score function**:

$$\frac{\nabla_\theta \pi(a|S_t, \theta)}{\pi(a|S_t, \theta))} = \nabla_\theta log \pi(a|S_t, \theta)$$

- Substituting the score function gives the gradient:

$$\nabla_\theta \pi(a|S_t, \theta) = \pi(a|S_t, \theta) \nabla_\theta log \pi(a|S_t, \theta)$$

Reinforce Algorithm

Direct application of policy gradient theorem: **Reinforce algorithm**

- Stochastic gradient ascent method
- Basic algorithm:
  1. Unbiased estimate of v(s) with Monte Carlo estimation
  2. Update the parameters:

     $$\theta_{t+1} = \theta_t + \alpha \, \nabla_\theta log\pi(a|S_t, \theta) \, v_t(s)$$

     where α is the learning rate
  3. Repeat 1. and 2.
- But, policy gradient has high variance

Stochastic Policies

Deterministic vs. **stochastic policy**

- Deterministic policy maximizes state-value or action-value, e.g:
  - Max_a Q(s,a)
  - Or, ε-greedy
- **Stochastic policy** uses probabilistic actions
- Stochastic policy useful when uncertainty in the optimal action:
  - Data from unreliable or inaccurate sensors: poor GPS signal
  - Response to actions uncertain: Self-driving car on ice

Stochastic Policies

Stochastic policy for **discrete actions**

- Use **softmax action preferences**:

$$\pi(a|s,\theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}}$$

- For **linear function approximation**, $\phi(s,a)\ \theta$:

$$\pi(a|s,\theta) \propto e^{\phi(s,a)^T\ \theta}$$

- The score function is then:

$$\nabla_\theta \pi(a|S_t,\theta) = \phi(s,a) - \mathbb{E}_{\pi_\theta}\left[\phi(s,\cdot)\right]$$

Reducing Variance with a Critic

Direct application of policy gradient has **high variance**; e.g. Reinforce

- Can use **Actor-Critic method** to reduce variance
  - **Critic** evaluates the policy
  - Critic uses approximate action-value parameterized by weight vector, **w**

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

  - **Actor** determines action policy
  - Actor parameterized by vector, θ, updated using critic

Reducing Variance with a Critic

Direct application of policy gradient has **high variance**; e.g. Reinforce

- Can use **Actor-Critic method** to reduce variance
  - **Critic** evaluates the policy
  - Critic uses approximate action-value parameterized by weight vector, **w**

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

  - **Actor** determines action policy
  - Actor parameterized by vector, θ, updated using critic

Reducing Variance with a Critic

Actor-Critic uses **approximate policy gradient**

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}\left[\nabla_\theta\ log\ \pi_\theta(s,a)\ Q_w(s,a)\right]$$

- With the parameter update:

$$\Delta\theta = \alpha\ \nabla_\theta\ log\ \pi_\theta(s,a)\ Q_w(s,a)$$

- Methods to approximate $Q_w(s,a)$:
  - Monte Carlo action-value evaluation
  - TD bootstrap methods
  - Least squares minimization function approximation

Bias in Actor-Critic Methods

Actor-Critic methods can introduce considerable **bias** in the solution

- **Bias** leads to **poor convergence** and **sub-optimal solutions**
- Meet two criteria to limit bias:
  - Value function must be **compatible** with the policy:

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

  - Parameterized value function **minimizes mean squared error**:

$$\epsilon = \mathbb{E}_{\pi_\theta}\left[\left(Q_{\pi_\theta}(s,a) - Q_w(s,a)\right)^2\right]$$

- An exact policy gradient that meets both criteria

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \, log \, \pi_\theta(s,a) \, Q_w(s,a)\right]$$

Learning the Critic Function

How to update the critic?

- Use gradient asset to update parameter vector **w**:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\, \delta_t\, \nabla_w\, Q_w(s, a)$$

where,

$\delta_t = R_{t+1} + \gamma Q(S_{t+1}, a) - Q(S_t, A_t)$ is the TD error

$\nabla_w\, Q_w(s, a)$ is the gradient of $Q_w(s, a)$ with respect to **w**.

Learning the Critic Function

Example of computing $\nabla_w \; Q_w(s,a)$:

- Start with a linear function approximation:

$$Q_w(s,a) = \phi(s,a)^T \mathbf{w}$$

where, $\phi(s,a)$ are the basis functions

- Since the approximation is linear in **w**:

$$\nabla_w \; Q_w(s,a) = \phi(s,a)$$

Advantage Actor-Critic: A2C

Reduce variance using **baseline function**

- Easy method to reduce variance, minimize range of values by introducing baseline function
- The expected value of the policy gradient with baseline:

$$
\begin{aligned}
\nabla_\theta J(\theta) &\propto \sum_s \mu(s) \sum_a \nabla_\theta \pi(a|S_t, \theta) \Big(q_\pi(s, q) - b(s) \Big) \\
&= \mathbb{E}_{\pi_\theta} \Big[\sum_a \nabla_\theta \pi(a|S_t, \theta) \Big(q_\pi(s, q) - b(s) \Big) \Big]
\end{aligned}
$$

Advantage Actor-Critic: A2C

Baseline function does not change expectation of the gradient

- Sum of the baseline component:

$$\sum_a \nabla_\theta \pi(a|S_t, \theta)\, b(s) = b(s) \sum_a \nabla_\theta \pi(a|S_t, \theta)$$

- But, the sum of probabilities over all actions is just 1, so:

$$\sum_a \nabla_\theta \pi(a|S_t, \theta)\, b(s) = \, b(s)\, \nabla_\theta 1 = 0$$

Advantage Actor-Critic: A2C

Need to choose a baseline function

- Use the **state-value function** as baseline
- Gives the **advantage function**:

$$A_{\pi_\theta}(s, a) = Q_{\pi_\theta}(s, a) - V_{\pi_\theta}(s)$$

- At convergence the action-value and state-value are equal so $A_{\pi_\theta}(s, a)$ converges to 0
- The lower variance policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \, log \; \pi_\theta(s, a) \, A_{\pi_\theta}(s, a)\right]$$

Summary of Actor-Critic Updates

Methods to find the policy parameter update

$\Delta\theta = \alpha \nabla_\theta log\pi(a|S_t, \theta)\, v_t(s)$ Reinforce

$\Delta\theta = \alpha \nabla_\theta \, log \, \pi_\theta(s, a)\, Q_w(s, a)$ Q Actor-Critic

$\Delta\theta = \alpha \nabla_\theta \, log \, \pi_\theta(s, a)\, A_{\pi_\theta}(s, a)$ Advantage Actor-Critic