Policy Gradient Method

Action-value Methods : learned the values of actions and then selected actions based on their estimated action values

Learn a parameterized policy : select actions without consulting a value function

$\theta \in \mathbb{R}^{d'}$: policy's parameter vector

$$\pi(a|s, \theta) = \Pr\{A_t = a \mid S_t = s, \theta_t = \theta\}$$

$W \in \mathbb{R}^{d}$: value function's weight vector

$\hat{V}(s, w)$

"Learn the policy parameter based on the gradient of some scalar performance measure $J(\theta)$ w.r.t. the policy parameter

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

$\widehat{\nabla J(\theta_t)} \in \mathbb{R}^d$: Stochastic estimate whose expectation approximates the gradient of the performance measure w.r.t. its argument θ_t

- learn value function : action-value method
- Learn policy : policy gradient method
- Learn value function & policy : actor-critic method

softmax in action preference :

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

$$h(s,a,\theta) = \theta^T x(s,a)$$

① deterministic policy,

② action selection with arbitrary probabilities

The Policy Gradient Theorem

"How performance is affected by the policy parameter that does not involve derivatives of the state distribution"

Value of the start state under the parameterized policy

performance: $J(\theta) \doteq V_{\pi_\theta}(s_0)$ average reward rate

$$\nabla J(\theta) = \sum_{s'} \eta(s') \sum_{s} \mu(s) \sum_{a} \nabla \pi(a|s)\, q_\pi(s,a)$$

$$\propto \sum_{s} \mu(s) \sum_{a} q_\pi(s,a) \nabla \pi(a|s,\theta)$$

proportional to

REINFORCE: MC Policy Gradient

$$\nabla J(\theta) \propto \sum_{s} \mu(s) \sum_{a} q_\pi(s,a) \nabla \pi(a|s,\theta)$$

$$= \mathbb{E}\left[\sum_{a} q_\pi(S_t, a) \nabla_\pi(a|S_t, \theta) \right]$$

$$= \mathbb{E}_\pi\left[G_t \frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)} \right]$$

$$\theta_{t+1} \doteq \theta_t + \alpha \sum_a \hat{q}(S_t, a, w) \nabla \pi(a|S_t, \theta)$$

*

$$\boxed{\theta_{t+1} \doteq \theta_t + \alpha \boxed{G_t} \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)}}$$

$\nabla \ln \pi(A_t|S_t, \theta_t)$

"use the complete return from time t, which includes all future rewards up until the end of the episode."

REINFORCE: MC Policy-gradient Control (episodic) for π:

$$G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

$$\theta \leftarrow \theta + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \theta)$$

REINFORCE Proof:

$$\nabla J(\theta) = \mathbb{E}_\pi \left[\sum_a \pi(a|S_t, \theta)\, q_\pi(S_t, a) \frac{\nabla \pi(a|S_t, \theta)}{\pi(a|S_t, \theta)} \right]$$

$$= \mathbb{E}_\pi \left[q_\pi(S_t, A_t) \frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)} \right]$$

$$= \mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)} \right]$$

REINFORCE with Baseline (unbiased)

"include a comparison of the action value to an arbitrary baseline $b(s)$"

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a \left(q_\pi(s,a) - b(s) \right) \nabla \pi(a|s,\theta)$$

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

$$\theta_{t+1} \doteq \theta_t + \alpha \left(\boxed{G_t} - b(S_t) \right) \frac{\nabla \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

$$\begin{cases} G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \\ \delta \leftarrow G - \hat{v}(S_t, w) \\ w \leftarrow w + \alpha^w \delta \nabla \hat{v}(S_t, w) \\ \theta \leftarrow \theta + \alpha^\theta \gamma^t \delta \nabla \ln \pi(A_t|S_t,\theta) \end{cases}$$

$$\alpha^w = \frac{0.1}{E\left[\| \nabla \hat{v}(S_t, w) \|_\mu^2 \right]}$$

Actor-critic Methods

$$\theta_{t+1} \doteq \theta_t + \alpha\left(G_{t:t+1} - \hat{v}(S_t, w)\right)\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

$$= \theta_t + \alpha\left(\boxed{R_{t+1} + \gamma\hat{v}(S_{t+1}, w) - \hat{v}(S_t, w)}\right)\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

$$= \theta_t + \alpha\delta\frac{\nabla\pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$$

One-step actor-critic methods replace the full return of REINFORCE with the one-step return (and use a learned state-value function as the baseline)