

Policy Gradient Methods

esgl Hu

November 26, 2017

1 preface

Notations & Definitions

The methods have learned the values of actions and then selected actions based on their estimated action values, their policies would not even exist without the action-value estimates.

The methods have learned a *parameterized policy* that can select actions without consulting a value function. A value function may still be used to *learn* the policy weights, but is not required for action selection.

The notation $\theta \in \mathbb{R}^n$ is used for the primary learned weight vector, $\pi(a|s, \theta) \doteq \Pr\{A_t = a | S_t = s, \theta_t = \theta\}$ for the probability that action a is taken at time t given that the agent is in state s at time t weight vector θ .

These methods seek to maximize performance $\eta(\theta)$, so their updates approximate gradient ascent in η :

$$\theta_{t+1} \doteq \theta_t + \alpha \widehat{\nabla(\eta(\theta_t))} \quad (1)$$

where $\widehat{\nabla(\eta(\theta_t))}$ is a stochastic estimate whose expectation approximates the gradients of the performance measure $\eta(\theta_t)$ with respect to its argument θ_t

Advantages & Disadvantages

Advantages

- Better convergence properties
- Effective in high-dimensional or continuous action spaces
- Can learn stochastic policies

Disadvantages

- Typically converge to a local rather than global optimum
- Evaluating a policy is typically inefficient and high variance

Definition of Policy Gradient

We let τ denote a state-action sequence $\{s_0, a_0, \dots, s_H, a_H\}$. We overload notation: $R(\tau) = \sum_{t=0}^H R(s_t, a_t)$.

$$\eta(\theta) = \mathbb{E}\left[\sum_{t=0}^H R(s_t, a_t), \pi_\theta\right] = \sum_{\tau} P(\tau; \theta) R(\tau) \quad (2)$$

In our new notation, our goal is to find θ :

$$\max_{\theta} \eta(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \quad (3)$$

Algorithm 1 REINFORCE: A Monte-Carlo Policy-Gradient Method (episodic)

Input: a differentiable policy parameterization $\pi(a|s, \theta)$, $\forall a \in A, s \in S, \theta \in R^n$

Initialize policy weights θ

while True **do**

 Generate an episode $\{S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T, S_T\}$ follow $\pi(\cdot|\cdot, \theta)$

for each step of the episode $t = 0, 1, \dots, T-1$ **do**

$G_t \leftarrow$ return from step t

$\theta \leftarrow \theta + \alpha \gamma^t G_t \nabla_{\theta} \log \pi(A_t|S_t, \theta)$

end for

end while
