RL: Policy Search

The Big Picture

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover
AutoML.org

Policy-Based Reinforcement Learning

- In the last lecture we approximated the value or action-value function using parameters $\mathbf{w}$,

$$V_{\mathbf{w}}(s) \approx V^{\pi}(s)$$

$$Q_{\mathbf{w}}(s, a) \approx Q^{\pi}(s, a)$$

- A policy was generated directly from the value function
  - e.g., using ϵ-greedy
- Now, we will directly parametrize the policy, and will typically use θ to show parameterization:

$$\pi_{\theta}(s, a) = \mathbb{P}[a \mid s; \theta]$$

- Goal is to find a policy π with the highest value function V^{π}
- We will focus again on model-free reinforcement learning

Automated Machine Learning Hannover

Value-Based and Policy-Based RL

- Value-based
  - Learn Value function
  - implicit policy (e.g., ϵ-greedy)
- Policy-based
  - No explicit value function
  - learnt policy
- Actor-Critic
  - Learn Value Function
  - Learn Policy

Types of Policies to Search Over

- So far have focused on deterministic policies
- Now we are thinking about direct policy search in RL, will focus heavily on stochastic policies

Example: Rock-Paper-Scissors

- Two-player game of rock-paper-scissors
  - Scissors beats paper
  - Rock beats scissors
  - Paper beats rock
- Let state be history of prior actions (rock, paper and scissors) and if won or lost
- Is deterministic policy optimal? Why or why not?

Automated
Machine Learning
Hannover

Example: Rock-Paper-Scissors

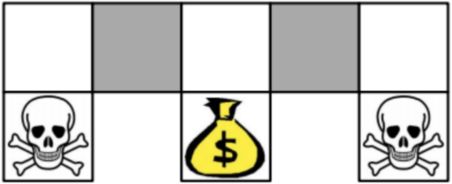
- Two-player game of rock-paper-scissors
  - Scissors beats paper
  - Rock beats scissors
  - Paper beats rock
- Let state be history of prior actions (rock, paper and scissors) and if won or lost
- Is deterministic policy optimal? Why or why not?

$\rightsquigarrow$ stochastic (random) policy is the Nash equilibrium

Automated Machine Learning Hannover

AutoML.org

Example: Aliased Gridword (1)



- Consider features of the following form (for all N, E, S, W)

$$\phi(s,a) = 1(\text{s="wall to N", a = "move E"})$$

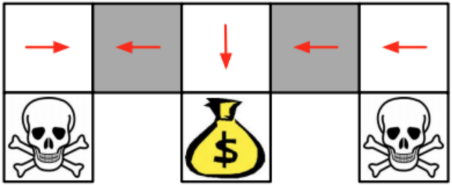
  - State representation is not Markov
  - The agent cannot differentiate the gray states
- Compare value-based RL, using an approximate value function

$$Q_\theta(s,a) = f(\phi(s,a);\theta)$$

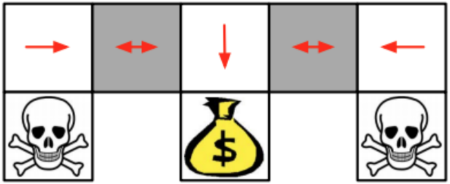
- To policy-based RL, using a parametrized policy

$$\pi_\theta(s,a) = g(\phi(s,a);\theta)$$

Example: Aliased Gridworld (2)



- Under aliasing, an optimal deterministic policy will either
  - Move W in both gray states
  - Move E in both gray states
- Either way, it can get stuck and never reach the money
- Value-based RL learns a near-deterministic policy
- So it will traverse the corridor for a long time



- An optimal stochastic policy will randomly move E or W in grey states

$$\pi_\theta(\text{wall to N and S, move E}) = 0.5$$

$$\pi_\theta(\text{wall to N and S, move W}) = 0.5$$

- It will reach the goal state in a few steps with high probability
- Policy-based RL can learn the optimal stochastic policy

Policy Objective Functions

- Goal: given a policy $\pi_\theta(s, a)$ with parameters θ, find best θ^*
- But how do we measure the quality for a policy π_θ
- In episodic environments, we can use policy value at start state $V(s_0, \theta)$
- For simplicity, we will mostly discuss the episodic case

Automated Machine Learning Hannover