

EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

Office: Olin 516

Office hours : T 4-5:30 or by appointment

Recap

- In general reinforcement learning the agent has to learn about the ____.
- What is “passive” RL?
- What does adaptive DP do?
- How do we estimate T and R in adaptive DP?
- What is model free RL?
- How can we use Monte Carlo for model free RL?
- Passive RL with Monte Carlo does not use the ____ ____.
- What is the temporal difference error?
- Why does TD learning work?
- What is “active” RL?
- We can try to extend ADP by starting with a random policy and passing the estimated T and R values to VI, but this does not work. Why?
- How does ϵ -greedy exploration work? Optimistic exploration? Boltzmann exploration?
- What is the property that is needed to learn the optimal policy?
- What is the correct way to extend ADP to learn the optimal policy?

Temporal Difference Learning

- Start with an arbitrary value function V_0
- Follow given policy
- For each observed $(s, \pi(s), s')$, do

$$V_{i+1}^{\pi}(s) \leftarrow V_i^{\pi}(s) + \alpha \left[R(s, \pi(s)) + \gamma V_i^{\pi}(s') - V_i^{\pi}(s) \right]$$

- Until $|V_{i+1}^{\pi}(s) - V_i^{\pi}(s)|$ is very small

Model Free Reinforcement Learning

- Start with an arbitrary value function V_0
- Follow **greedy policy with exploration**
- For each observed $(s, \pi(s), s')$, do

$$V_{i+1}^{\pi}(s) \leftarrow V_i^{\pi}(s) + \alpha \left[R(s, \pi(s)) + \gamma V_i^{\pi}(s') - V_i^{\pi}(s) \right]$$

- Until $|V_{i+1}^{\pi}(s) - V_i^{\pi}(s)|$ is very small

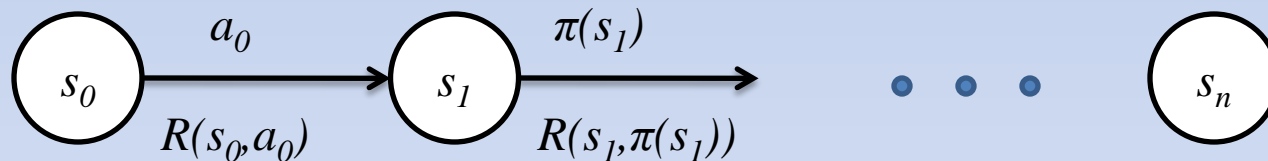
This is not model free, though it looks like it. In other words, it uses T and R implicitly somewhere. Where?

Getting the Policy from a Value Function

$$\pi^V(s) = \arg \max_a \left[R(s, a) + \gamma \sum_{s'} T(s, a, s') V(s') \right]$$

The Problem

- State values alone are not enough to find good actions!
- Instead, we need to estimate the *action-value* function, Q



$$Q^\pi(s_0, a_0) = \frac{1}{N} \sum_k \left[R(s_0, a_0) + \gamma R(s_1^k, \pi(s_1^k)) + \dots + \gamma^{n_k} R(s_{n_k}^k, \pi(s_{n_k}^k)) \right]$$

$$\pi(s) = \arg \max_a Q(s, a)$$

“Greedy” policy

Q -Learning

- The model-free counterpart of active ADP, based on TD-learning
- We define the **action-value function**, or Q function, $Q^\pi(s, a)$
 - The expected future reward for starting at s , taking action a , and following policy π thereafter

$$V^\pi(s) = E\left(\sum_t \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s\right)$$

$$Q^\pi(s, a) = E\left(\sum_t \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s, a_0 = a\right)$$

The $Q(s, a)$ function

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, a, s') V^\pi(s')$$

$$Q^\pi(s, a) = ??$$

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') Q^\pi(s', \pi(s'))$$

$$V^\pi(s) = Q(,) ??$$

$$V^\pi(s) = Q(s, \pi(s))$$

The $Q(s, a)$ function

$$V^{\pi^*}(s) = \max_a R(s, a) + \gamma \sum_{s'} T(s, a, s') V^{\pi^*}(s')$$

$$Q^{\pi^*}(s, a) = ??$$

$$Q^{\pi^*}(s, a) = R(s, a) + \gamma \max_{a'} \sum_{s'} T(s, a, s') Q^{\pi^*}(s', a')$$

$$\pi(s) = Q(,) ??$$

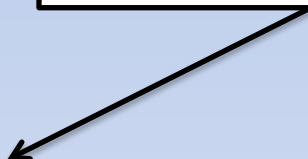
$$\pi(s) = \arg \max_a Q(s, a)$$

“Greedy” policy

Q-Learning

- Start with an arbitrary Q function Q_0
- Follow greedy policy with exploration
- For each observed (s, a, s') , do

Temporal difference error
for the Q-function



$$Q_{i+1}(s, a) \leftarrow Q_i(s, a) + \alpha \left[R(s, a) + \gamma \max_{a'} Q_i(s', a') - Q_i(s, a) \right]$$

- Until $|Q_{i+1}(s, a) - Q_i(s, a)|$ is small enough

On-policy vs. Off-policy Learning

- Notice that, due to exploration, sometimes Q-learning will back up action values that aren't actually exercised
 - This is called “off-policy” learning
 - An alternative algorithm, called SARSA, performs “on-policy” learning---it only backs up transitions that are actually taken

SARSA

- Start with an arbitrary Q function Q_0
- Follow greedy policy π with GLIE exploration
- For each observed (s, a, s') , do

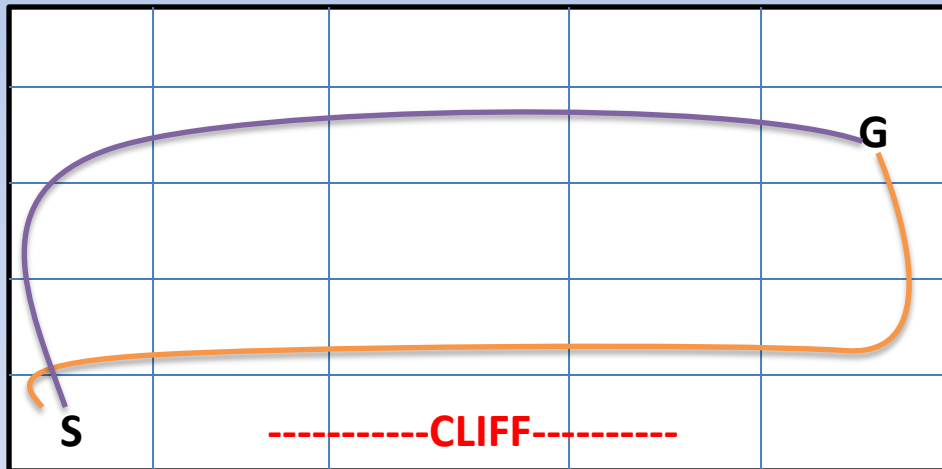
$$Q_{i+1}(s, a) \leftarrow Q_i(s, a) + \alpha [R(s, a) + \gamma Q_i(s', a') - Q_i(s, a)]$$

Only change: a' is the actual action taken in s'

- Until $|Q_{i+1}(s, a) - Q_i(s, a)|$ is small enough

Difference between on and off policy

- Cliff walking



SARSA chooses the “safer path” more often --- “falls off” less often than Q-learning

Algorithm Comparison

Have T and R ?	Task?	Model-based/free?	Method
Yes	Evaluate Policy	N/A	Solve Bellman Equations
Yes	Find Optimal Policy	N/A	Value/Policy Iteration
No	Evaluate Policy	Model-based	Adaptive DP
No	Evaluate Policy	Model-free	TD-learning
No	Find Optimal Policy	Model-based	“Active” Adaptive DP
No	Find Optimal Policy	Model-free	Q-learning

Model-based and model-free RL

- Given an RL problem, which should we use?
- Model-free methods are more flexible and storage efficient, but slower
- In some RL problems, a reasonable model can be quickly acquired; here model-based methods work well
- Interestingly, no difference in asymptotic rate of convergence between the two with respect to number of samples from environment, in the worst case