Quiz 8

Friday, May 17, 2024 7:48 AM

1 of 10

What is the main difference between **passive reinforcement learning** and evaluating a policy π by **solving an MDP** (Markov Decision Process) model?

A. There is no difference because passive reinforcement learning is also to evaluate a policy π.

B. Passive reinforcement learning can find the optimal policy.

C. In passive reinforcement learning, the transition model and the reward function are unknown.

D. Passive reinforcement learning can learn the action-value function $Q(s,a)$ while Policy Evaluation with MDP can learn the state-value function $V(s)$.

SUBMIT ANSWER

2 of 10

Initialize:
$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
Append G to $Returns(S_t)$
$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Zoom

What does the following algorithm do?

A. Estimating a state-value function for a policy π.

B. Estimating an action-value function for a policy π.

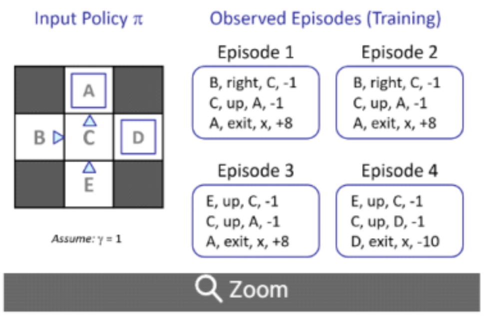
C. Estimating the optimal state-value function.

D. Estimating the optimal action-value function.

SUBMIT ANSWER

Initialize:
$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
Append G to $Returns(S_t)$
$V(S_t) \leftarrow \text{average}(Returns(S_t))$



Given the following environment with five states A, B, C, D, and E where A and D are two terminal states.

Let's consider the following policy π: π(A)=exit, π(B)=right, π(C)=up, π(D)=exit, π(E)=up.

We use the **Monte Carlo Direct Evaluation** method to evaluate the policy π. We obtain **four episodes** as in the below picture.

Assuming the discount factor γ=1, **compute the expected utility of the five states V(A)=?, V(B)=?, V(C)=?, V(D)=?, V(E)=?**

A V(A) = 8, V(B) = 6, V(C)=2, V(D) = -10, V(E) = -6

B V(A) = 8, V(B) = 6, V(C)=2.5, V(D) = -10, V(E) = -3

C V(A) = 6, V(B) = 3, V(C)=2.5, V(D) = -2.5, V(E) = -1.5

D V(A) = 8, V(B) = 6, V(C)=2, V(D) = -10, V(E) = -6

SUBMIT ANSWER

Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    $S \leftarrow S'$
  until S is terminal

Zoom

What is the following algorithm?

A A temporal difference learning algorithm to estimate the state-value function for a policy π.

B A Monte Carlo learning algorithm to estimate the state-value function for a policy π.

C SARSA algorithm for estimating the optimal state-value function.

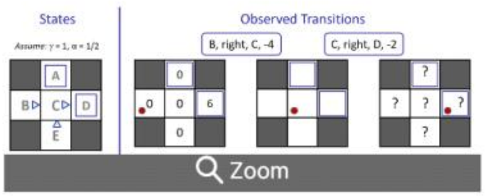
D SARSA algorithm for estimating the optimal action-value function.

SUBMIT ANSWER

Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

5 of 10



Given the following environment with five states A, B, C, D, and E where A and D are two terminal states.
Let's consider the following
policy π: π(A)=exit, π(B)=right, π(C)=right, π(D)=exit, π(E)=up.

We use the **Temporal Difference Learning** method to evaluate the policy π.

We have two transitions as in the below picture. (B,right,C,-4) gives us a reward of -4. (C, right, D,-2) gives us a reward of -2.

Assuming the discount factor gamma=1, learning rate alpha=1/2, **compute the expected utility of the five states after the two transitions V(A)=?, V(B)=?, V(C)=?, V(D)=?, V(E)=?**

A. V(A)=0, V(B)=-2, V(C)=2, V(D)=6, V(E)=0

B. V(A)=0, V(B)=-2, V(C)=-2, V(D)=3, V(E)=0

C. V(A)=-2, V(B)=2, V(C)=2, V(D)=6, V(E)=0

D. V(A)=0, V(B)=2, V(C)=-2, V(D)=6, V(E)=0

6 of 10

What is the key difference between **passive** reinforcement learning and **active** reinforcement learning?

A. Passive RL is to learn the value function for a fixed policy π while active RL is to learn the optimal policy/value function.

B. The transition function $P(s'|s, a)$ is unknown in active RL.

C. The reward function $R(s, a, s')$ is unknown in active RL.

D. We can employ the Value Iteration/Policy Iteration algorithms to solve active RL.

SUBMIT ANSWER

Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

Zoom

How should we change the following Reinforcement Learning algorithm so that we can estimate an optimal value function?

A. Estimating the action-value function $Q(S_t, A_t)$ instead of the state-value function $V(S_t)$.

B. Performing one-step lookahead to improve the current estimated state-value function $V(S_t) \leftarrow \max_a P(S_{t+1}|S_t, A_t)[R(S_t, A_t, S_{t+1}) + \gamma V(S_{t+1})]$

C. Performing policy improvement with respect to the current estimated action-value function, i.e., $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$.

D. Choose $S_0 \in S$, and $A_0 \in \mathrm{A}(S_0)$ randomly such that all pairs have probability > 0 .

E. A, B, and C need to be done.

F. A, C, and D need to be done.

G. B and D need to be done.

H. A, B, C, and D need to be done.

8 of 10

What is the $Q(s, a)$ update rule in Q-Learning? α is the learning rate, γ is the discount factor.

A. $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma Q(s', a'))$

B. $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s', a'))$

C. $Q(s,a) \leftarrow \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s,a))$

D. $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s', a') - Q(s,a))$

E. All are correct.

SUBMIT ANSWER

A is the set of actions a's we can perform at each state $s \in S$. Let $n = |A|$ be the number of actions.

An ε-greedy policy would choose a random action with probability ε.

Regarding an ε-greedy policy with respect to a action-value function $Q(s, a)$, in each step of **Q-Learning with ε-greedy exploration**, what is the probability that the action with the maximum $Q(s, a)$ value would be sampled?

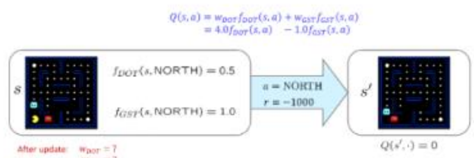
A $\frac{\varepsilon}{n}$

B ε

C $1 - \varepsilon$

D $1 - \varepsilon + \frac{\varepsilon}{n}$

SUBMIT ANSWER



Zoom

We are using **Approximate Q-Learning** with linear Q-value approximation for a basic Pacman game with two features: the reciprocal distance to the nearest food dot f_{DOT} (with weight w_{DOT})and the reciprocal distance to the nearest ghost f_{GST}(with weight w_{GST}).

Let's consider the current state s and action NORTH (going up) in the picture below. After Pacman performs the action NORTH, the blue ghost attacks Pacman, the obtained reward is -1000 in this transition, and the game ends.

Question: Using this transition, how the weights w_{DOT} and w_{GST} are updated?

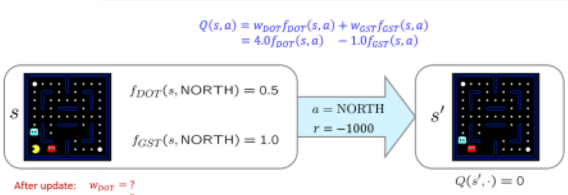
Note: discount factor gamma γ=1, and learning rate alpha $\alpha = 0.001$.

A $w_{DOT} \approx -3.5, w_{GST} \approx 2$

B $w_{DOT} \approx 3.5, w_{GST} \approx -2$

C $w_{DOT} \approx 3.5, w_{GST} \approx -10$

D $w_{DOT} \approx -3.5, w_{GST} \approx 10$



Finished!

Score: 9/10

Percent: 90%

OK