Reinforcement Learning CT 1 2025

Draft saved

* Indicates required question

Given below is equation of

$$V_{k+1}(s) = \sum_a \pi(s,a) \sum_{k'} P^a_{ss'} \left[r^a_{ss'} + \gamma V_k(s') \right]$$

- ( ) Policy iteration
- ( ) Policy evaluation
- ( ) Policy improvement
- (•) Value iteration

Clear selection

Given below is an equation of

$$\arg\max_a \sum_{s'} P^a_{ss'} \left[r^a_{ss'} + \gamma V^\pi(s') \right]$$

- ( ) Policy evaluation
- (•) Policy improvement
- ( ) Value iteration
- ( ) Policy iteration

Clear selection

Request edit access

Reinforcement learning is about

- [ ] transitions and rewards usually not available
- [x] how to change the policy based on experience
- [x] how to explore the environment
- [ ] Semi supervised learning

Discounted rewards is

- ( ) “game over” after N steps
- ( ) V(s0, s1, …) = r(s0) + r(s1) + r(s2) + …
- (•) V(s0, s1, …) = r(s0) + γ*r(s1) + γ2*r(s2) + …
- ( ) None of above

Clear selection

Monte Carlo methods

- (•) don’t need full knowledge of environment
- ( ) Defined for continuous tasks
- ( ) Summing sample patters
- ( ) V power pie enough for policy improvement

Clear selection

Which is true in monte carlo method

- [x] don’t need full knowledge of environment
- [ ] similar to Dynamic Programming
- [x] averaging sample returns
- [ ] need model of environment

Which of the following is not a useful way to approach a standard multi-armed bandit problem with n arms? Assume bandits are stationary.

- ( ) How can I ensure the best action is the one which is mostly selected as time tends to infinity?”
- ( ) “How can I ensure the total regret as time tends to infinity is minimal?”
- ( ) “How can I ensure an arm which has an expected reward within a certain threshold of the optimal arm is chosen with a probability above a certain threshold?”
- (•) “How can I ensure that when given any 2 arms, I can select the arm with a higher expected return with a probability above a certain threshold?”

Clear selection

Which of the following is/are correct and valid reasons to consider sampling actions from a softmax distribution instead of using an ϵ-greedy approach?
i. Softmax exploration makes the probability of picking an action proportional to the actionvalue estimates. By doing so, it avoids wasting time exploring obviously 'bad' actions.
ii. We do not need to worry about decaying exploration slowly like we do in the ϵ-greedy case. Softmax exploration gives us asymptotic correctness even for a sharp decrease in temperature.
iii. It helps us differentiate between actions with action-value estimates (Q values) that are very close to the action with maximum Q value.

Which of the above statements is/are correct?

- ○ i, ii, iii
- ○ only iii
- ○ only i
- ○ i, ii
- ◉ i, iii

Clear selection

We need 8 rounds of median-elimination to get an (ϵ,δ) − PAC arm. Approximately how many samples would have been required using the naive (ϵ,δ) − PAC algorithm given (ϵ,δ) = (1/2, 1/e) ? (Choose the value closest to the correct answer)

- ◉ 15000
- ○ 10000
- ○ 500
- ○ 20000

Clear selection

Suppose we are facing a non-stationary bandit problem. We want to use posterior sampling for picking the correct arm. What is the likely change that needs to be done to the algorithm so that it can adapt to non-stationarity?

- ○ Update the posterior rarely.
- ○ Randomly shift the posterior drastically from time to time.
- ◉ Keep adding a slight noise to the posterior to prevent its variance from going down quickly.
- ○ No change is required.

Clear selection

Which of the following statements is NOT true about Thompson Sampling or Posterior Sampling?

- ○ After each sample is drawn, the q_* distribution for that sampled arm is updated to be closer to the true distribution.
- ○ Thompson sampling has been shown to generally give better regret bounds than UCB.
- ○ In Thompson sampling, we do not need to eliminate arms each round to get good sample complexity.
- ◉ The algorithm requires that we use Gaussian priors to represent distributions over q_* values for each arm.
- ○ Option 5

Clear selection

In many supervised machine learning algorithms, such as neural networks, we rely on the gradient descent technique. However, in the policy gradient approach to bandit problems, we made use of gradient ascent. This discrepancy can mainly be attributed to the differences in

- ☑ the objectives of the learning tasks
- ☐ the parameters of the functions whose gradient are being calculated
- ☐ the nature of the feedback received by the algorithms

The actions in contextual bandits do not determine the next state, but typically do in full RL problems. True or false?

- (●) True
- ( ) False

Clear selection

Let's assume for some full RL problem we are acting according to a policy π. At some time t, we are in a state s where we took action a1. After few time steps, at time t', the same state s was reached where we performed an action a2(≠a1). Which of the following statements is true? π is

- ( ) is definitely a stationary policy
- ( ) is definitely a non-stationary policy
- (●) can be stationary or non-stationary

Clear selection

State True/False
The state transition graph for any MDP is a directed acyclic graph.

- ( ) True
- (●) False

Clear selection

Which of the following is a benefit of using RL algorithms for solving MDPs?

- ○ They do not require the state of the agent for solving a MDP.
- ○ They do not require the action taken by the agent for solving a MDP.
- ◉ They do not require the state transition probability matrix for solving a MDP.
- ○ They do not require the reward signal for solving a MDP.

Clear selection

Consider an MDP with 3 states A, B, C. From each state, we can go to either of the two states, i.e, from state A, we can perform 2 actions, that lead to state B and C respectively. The rewards for all the transitions are: r(A, B) = 2 (reward if we go from A to B), r(B, A) = 5, r(B, C) = 7, r(C, B) = 10, r(A, C) = 1, r(C, A) = 12. The discount factor is 0.7. Find the value function for the policy given by: $\pi(A)=C$(if we are in state A, we choose the action to go to C),$\pi(B)=A$ and $\pi(C)=B([v\pi(A),v\pi(B),v\pi(C)])$

- ○ [10.2, 16.7, 20.2]
- ○ [14.2, 16.5, 15.1]
- ◉ [15.9, 16.1, 21.3]
- ○ [12.2, 6.2, 14.5]

Clear selection

Consider Monte-Carlo approach for policy evaluation. Suppose the states are S1,S2,S3,S4,S5,S6 and terminal_state. You sample one trajectory as follows - S1→S5→S4→S6→ terminal_state.
Which among the following states can be updated from this sample?

- ☑ S1
- ☐ S2
- ☑ S6
- ☑ S4

Request edit access

Which of the following statements are **FALSE** about solving MDPs using dynamic programming?

- [ ] f the state space is large or computation power is limited, it is preferred to update only some states through random sampling or selecting states seen in trajectories.
- [x] Knowledge of transition probabilities is not necessary for solving MDPs using dynamic programming.
- [x] Methods that update only a subset of states at a time guarantee performance equal to or better than classic DP.
- [ ] None of the above.

What is meant by ”off-policy” Monte Carlo value function evaluation?

- ( ) The policy being evaluated is the same as the policy used to generate samples.
- (•) The policy being evaluated is different from the policy used to generate samples.
- ( ) The policy being learnt is different from the policy used to generate samples.
- ( ) Option 4

Clear selection

Submit

Clear form