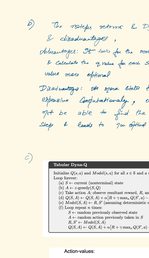
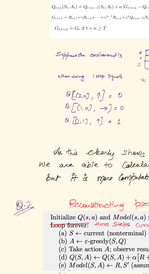
Q4

Tuesday, November 12, 2024 1:00 PM

Q.1

Q.2





Q.3

a) The Multi-Step bootstrapping method could also be used on the Dyna-Q.

Dyna-Q approximation would say on Section 6.5 Dyna, you write the Simulation Loop then to the Planning Step, whereas in the n-step bootstrapping we update in Dyna-Q. So the update are done in the Dyna-Q in the better than the n-step bootstrapping, as the updates state & values & explore both real & Simulation experience

b) The n-step return & Dyna have its advantages & disadvantages.

Advantages: It helps to the more accurate & frequently visited states & calculate the q-value for some state-action pair & to add give value more efficient

Disadvantages: the agent tends to be stuck in some suboptimal computationally, and might not be able to find the opt of the next in step & tends to the optimal Q value

c)

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in S$ and $a \in A(s)$

Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A, observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$

Action-value:

$$Q_{t+1}(S_t, A_t) = Q_{t+1-(S_t, A_t)} + \alpha_{t+1-(S_t, A_t)}[R_t - Q_{t+1-(S_t, A_t)}] \quad 0 \le t < T$$

$$Q_{t+1} = Q_{t+1} + \alpha_{t+1}[R_{t+1} + \gamma \max_a Q_{t+1}(S_{t+1}, a) - Q_{t+1}(S_t, A_t)] \quad \ge 1 \le t < T-1$$

$$Q_{t+1} = Q_t \text{ if } t = \ge T$$

Suppose environment

when using 1 step Dyna-Q

$Q[2,0], \uparrow] = 0$

$Q[(1,0), \rightarrow] = 0$

$Q[(1,1), \uparrow] = 1$

Step Dyna-Q

$Q[(1,0), \uparrow] = 0$

$Q[(1,1), \uparrow] = 0.95$

$Q[(2,0), \uparrow] = 0.95 \cdot \gamma^2 = 0.95$

In this clearly shows that with 2-step Dyna-Q we are able to calculate more optimal Q value but it is more computationally expensive

Q.2

Reconstructing baseline for Dyna Q+

Tabular Dyna-Q+

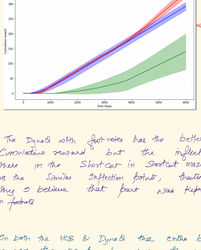
Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in S$ and $a \in A(s)$

Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment), τ
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action from all action
  - $R, S', \tau \leftarrow Model(S,A)$
  - $R \leftarrow R + \kappa\sqrt{\tau}$ (bonus reward ($\kappa \ll 1$))
  - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$

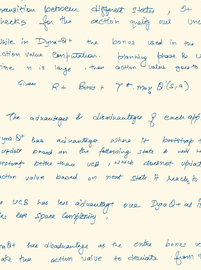
else

$Model(S,A) \leftarrow 0, S', 0$

Q.2.b



Blocking maze



Shortcut maze

The Dyna-Q with high rates has the better Cumulative reward but the exploration phase in the Shortcut maze has the similar exploration result, while Dyna-Q believes that first path kept in further

Q.3

a) In both the UCB & Dyna-Q the entire bonus encourages the agent to choose the actions that haven't been chosen in a long time, In UCB - K-arm bandit it doesn't transition between different states, It only checks for the action quality and Uncertainty

While in Dyna-Q the bonus used in the action value computation, planning phase & update time, n is large, max action value goes to infinity

Given $R \leftarrow Bonus + \gamma \max_a Q(s',a)$

b) The advantages & disadvantages of each approach

Dyna-Q has advantages where it bootstrap the update based on the following state & self service bootstrap better than UCB, which doesn't update action value based on next state it needs to

The UCB has less advantage over Dyna-Q as it takes less Space Complexity.

Dyna-Q has disadvantage as the entire bonus reward make the action value to deviate from the true value

UCB has disadvantage as it doesn't handle the transition well as it doesn't update action value

Q.4

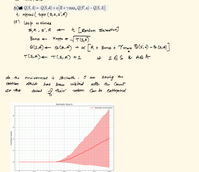
a) Stochastic environment,

Store Transition $T(s,a)$

Using these Transition τ' in the planning step

Tabular $Q(s,a)$ & Transition $T(s,a) \leftarrow 0$

$\varepsilon \in [0,1]$ & Number timing $n \in \mathbb{N}$

Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A, observe resultant reward, R, and state, S'
- (d) $T[S,A] \leftarrow$
- (e) $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_a Q(S',a) - Q(S,A)]$
- (f) $Model(S,A) \leftarrow R, S'$
- (g) Loop n times:
  - $R, S, \tau \leftarrow$ & (Random Element)
  - Bonus $\leftarrow$ Bonus $\times \sqrt{T[S,A]}$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \text{Bonus} + \gamma \max_a Q(S,a) - Q(S,A)]$
  - $T[S,A] \leftarrow T[S,A] + 1 \quad \forall \; S \in S \; \& \; A \in A$

As this environment is stochastic, it can change the action which has been visited with the Transition so the bonus of their action can be increased