CSC 449 Advanced Topics in Artificial Intelligence

Exam 2

Nov 30, 2022

Patrick McBride

90

Problem 1:

Consider the following pseudo-code for a faulty SARSA algorithm:

procedure SARSA(number of episodes $N \in \mathbb{N}$
discount factor $\lambda \in (0,1]$
learning rate $\alpha_n = \frac{1}{\log(n+1)}$)

Error 1

Initialize matrices $Q(s,a)$ and $n(s,a)$ to $0, \forall s,a$
for episode $k \in 1,2,3,\ldots,n$ **do**
$t \leftarrow 1$
Initialize s_1
Choose a_1 from a uniform distribution over the actions

Error 2

while Episode k is not finished **do**
Take action a_t: observe reward r_t and next state s_{t+1}
Choose a_{t+1} from s_{t+1} using μ_t: an ε-greedy policy with respect to Q
if The current state is terminal **then** ▷ *Compute target value*

$$y_t = 0$$

should be r

else

$$y_t = r_t + \max_a Q(s_{t+1}, a)$$

no max, and missing lamba

end if
$n(s_t,a_t) \leftarrow n(s_t,a_t) + 1$
Update Q function:

$$Q(s_{t+1},a_{t+1}) \leftarrow Q(s_t,a_t) - \alpha_{n(s_t,a_t)}(y_t - Q(s_t,a_t))$$

Error 3

$t \leftarrow t+1$
end while
end for
end procedure

Find all of the mistakes in the algorithm. Explain why they are mistakes, and correct them.

1) This adaptive learning rate is ok so long as log is base 2 since alpha must be bounded by (0,1]

2) Basic SARSA(λ) is an on-policy TD control algorithm, so the action a_1 must be chosen from s_1 using a policy derived from Q(ε - greedy) (AKA μ_t).

3) After confirming that the intended implementation for this pseudo-code is SARSA(λ). The update to Q should include eligibility traces. (Since we were not given a λ we will assume it is 1). So the Q update should be:

$\forall s \in S, a \in A(s)$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \, n(s_t, a_t) \, [y_t - Q(s_t, a_t)]$$

$$n(s_t, a_t) \leftarrow \lambda n(s_t, a_t)$$

Problem 2:

Your friend found a variant of SARSA which is defined through a sequence of policies π_t (where $t \geq 1$), and consists of just changing (in the previous algorithm after corrections) the way the target is computed. The target becomes

$$y_t = r_t + \lambda \sum \pi_t(a|s_{t+1}) Q(S_{t+1}, a),$$

where $\pi_t(a|s)$ is the probability that a is selected in state s under policy π_t .

A. What sequence of policies (π_t) should you choose so that the corresponding variant of SARSA is on-policy? This variant is called Expected SARSA.

For a RL algorithm to be on-policy it must follow the policy that it is learning when choosing actions. This means that for Expected SARSA the action that is taken must be that action with the highest value in the current state. The action taken will be decided by a deterministic policy that is ε-greedy with respect to the current Q function. In other words: $\forall t$, π_t will be replaced by π_{t+1} which is an ε-greedy policy with respect to the current Q function.

B. Consider an off-policy variant of SARSA corresponding to a stationary policy $\pi = \pi_t \forall t$. Under this algorithm, do the Q values converge? If so, what are the limiting Q values? Justify your answer.

If following π_t will result in eventually reaching the terminal state:

The Q values will converge. They will be limited by the Q value of the terminal state. This is because the Q value of the terminal state will never change as it is never left. Therefore all other states will have their Q values based off the Q value of the terminal state.

If following π_t will never result in reaching the terminal state:

The Q values will not converge since upon each update the will be driven further negative. As the number of iterations grows the Q values of the states that we visit will eventually be updated by $-\infty$ each time.